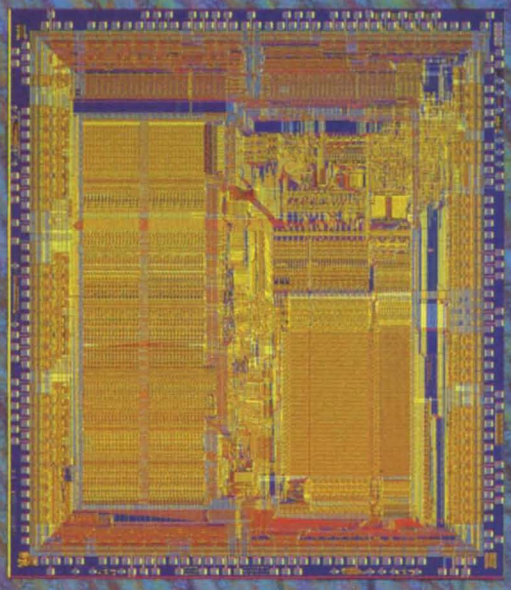
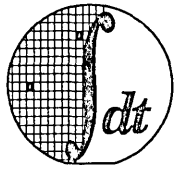
1991 DATABOOK

RISC



Integrated Device Technology, Inc.



Integrated Device Technology, Inc.

1991
RISC
DATA BOOK

3236 Scott Boulevard, Santa Clara, California 95054
Telephone: (408) 727-6116 • TWX: 910-338-2070 • FAX: (408) 492-8674
Printed in U.S.A.

GENERAL INFORMATION

CONTENTS OVERVIEW

Historically, Integrated Device Technology has presented our product offerings entirely under one cover. For ease of use for our customers, we have divided the products into four separate data books — Logic, Specialized Memory, RISC and Static RAM.

IDT's 1991 RISC Data Book is comprised of new and revised data sheets and application notes for the RISC and RISC Subsystem product lines. Also included is a current, complete packaging section for all IDT product groups. This section will be updated in each subsequent data book with the latest available packages.

The RISC Data Book's Table of Contents contains a listing of the products contained in the 1991 RISC Data Book, as well as those products which are contained in the other three data books. The numbering scheme is slightly different from the past. The number in the bottom center of the page denotes the section number and the sequence of the data sheet within that section, (i.e. 5.5 would be the fifth data sheet in the fifth section). The number in the lower right hand corner is the page number of that particular data sheet.

Integrated Device Technology, a recognized leader in high-speed CMOS technology, produces a broad line of products, enabling us to provide a complete CMOS solution to designers of high-performance digital systems. Our products include industry standard devices, as well as products with speed, lower power, package and/or architectural benefits that allow the designer to achieve significantly improved system performance.

Use this book to find ordering information: Start with the Ordering Information chart at the back of each data sheet or the Cross Reference Guides (in Section 1), along with the Package Outline Index (page 4.2), to compose the complete IDT part number. Reference data on our Technology Capabilities and Quality Commitments are included in separate sections (2 and 3, respectively).

Use this book to find product data: Start with the Table of Contents, organized by product line (page 1.3), or with the Numeric Table of Contents across all product lines (page 1.4). These indexes will direct you to the page on which the complete technical data sheet can be found. Data sheets may be of the following type:

ADVANCE INFORMATION — contain initial descriptions, subject to change, for products that are in development, including features and block diagrams.

PRELIMINARY — contain descriptions for products soon to be, or recently, released to production, including features, pinouts and block diagrams. Timing data are based on simulation or initial characterization and are subject to change upon full characterization.

FINAL — contain minimum and maximum limits specified over the complete supply and temperature range for full production devices.

New products, product performance enhancements, additional package types and new product families are being introduced frequently. Please contact your local IDT sales representative to determine the latest device specifications, package types and product availability.

LIFE SUPPORT POLICY

Integrated Device Technology's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of IDT.

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

1991 RISC DATA BOOK
SUMMARY TABLE OF CONTENTS

1

PAGE

SUMMARY TABLE OF CONTENTS (CONTINUED) BOOK

1991 RISC DATA BOOK
TABLE OF CONTENTS

1

PAGE

1990-91 LOGIC DATA BOOK

The following is a listing of the data sheets located in the 1990-91 Logic Data Book available under separate cover:

COMPLEX LOGIC PRODUCTS

STANDARD LOGIC PRODUCTS

STANDARD LOGIC PRODUCTS (CONTINUED)

1990-91 LOGIC DATA BOOK (CONTINUED)

APPLICATION AND TECHNICAL NOTES

Complex Logic Products Technical Notes

1990-91 LOGIC DATA BOOK (CONTINUED)

SPECIALIZED MEMORIES DATA BOOK

The following is a listing of the data sheets located in the 1990-91 Specialized Memories Data Book available under separate cover:

1

ECL PRODUCTS

FIFO PRODUCTS

FIFO PRODUCTS (CONTINUED)

SPECIALTY MEMORY PRODUCTS

1990-91 SPECIALIZED MEMORIES DATA BOOK (CONTINUED)

STATIC RAM DATA BOOK

The following is a listing of the data sheets located in the 1991 Static RAM Data Book available under separate cover:

STATIC RAM PRODUCTS

NUMERICAL TABLE OF CONTENTS

NUMERICAL TABLE OF CONTENTS (CONTINUED)

NUMERICAL TABLE OF CONTENTS (CONTINUED)

NUMERICAL TABLE OF CONTENTS (CONTINUED)

NUMERICAL TABLE OF CONTENTS (CONTINUED)

NUMERICAL TABLE OF CONTENTS (CONTINUED)

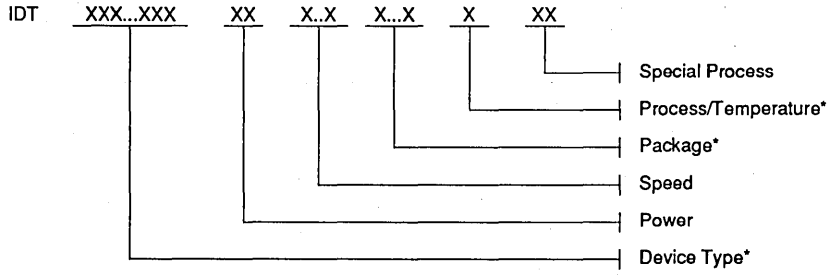
IDT PACKAGE MARKING DESCRIPTION

PART NUMBER DESCRIPTION

IDT's part number identifies the basic product, speed, power, package(s) available, operating temperature and processing grade. Each data sheet has a detailed description, using the part number, for ordering the proper product for the user's application. The part number is comprised of a series of alpha-numeric characters:

1. An "IDT" corporate identifier for Integrated Device Technology, Inc.
2. A basic device part number composed of alpha-numeric characters.
3. A device power identifier, composed of one or two alpha characters, is used to identify the power options. In most cases, the following alpha characters are used:
   "S" or "SA" is used for the standard product's power.
   "L" or "LA" is used for lower power than the standard product.
4. A device speed identifier, when applicable, is either alpha characters, such as "A" or "B", or numbers, such as 20 or 45. The speed units, depending on the product, are in nanoseconds or megahertz.
5. A package identifier, composed of one or two characters. The data sheet should be consulted to determine the packages available and the package identifiers for that particular product.
6. A temperature/process identifier. The product is available in either the commercial or military temperature range, processed to a commercial specification, or the product is available in the military temperature range with full compliance to MIL-STD-883. Many of IDT's products have burn-in included as part of the standard commercial process flow.
7. A special process identifier, composed of alpha characters, is used for products which require radiation enhancement (RE) or radiation tolerance (RT).

Example for Monolithic Devices:



* Field Identifier Applicable To All Products

2507 drw 01

ASSEMBLY LOCATION DESIGNATOR

IDT uses various locations for assembly. These are identified by an alpha character in the last letter of the date code marked on the package. Presently, the assembly location alpha character is as follows:

A = Anam, Korea
I = USA
P = Penang, Malaysia

MIL-STD-883C COMPLIANT DESIGNATOR

IDT ships military products which are compliant to the latest revision of MIL-STD-883C. Such products are identified by a "C" designation on the package. The location of this designator is specified by internal documentation at IDT.

IDT...LEADING THE CMOS FUTURE

A major revolution is taking place in the semiconductor industry today. A new technology is rapidly displacing older NMOS and bipolar technologies as the workhorse of the 80's and beyond. That technology is high-speed CMOS. Integrated Device Technology, a company totally predicated on and dedicated to implementing high-performance CMOS products, is on the leading edge of this dramatic change.

Beginning with the introduction of the industry's fastest CMOS 2K x 8 static RAM, IDT has grown into a company with multiple divisions producing a wide range of high-speed CMOS circuits that are, in almost every case, the fastest available. These advanced products are produced with IDT's proprietary CEMOS™ technology, a twin-well, dry-etched, stepper-aligned process utilizing progressively smaller dimensions.

From inception, IDT's product strategy has been to apply the advantages of it's extremely fast CEMOS technology to produce the integrated circuit elements required to implement high-performance digital systems. IDT's goal is to provide the circuits necessary to create systems which are far superior to previous generations in performance, reliability, cost weight and size. Many of the company's innovative product designs offer higher levels of integration, advanced architectures, higher density packaging and system enhancement features that are establishing tomorrow's industry standards. The company is committed to providing its customers with an ever-expanding series of these high-speed, lower-power IC solutions to system design needs.

IDT's commitment, however, extends beyond state-of-the-art technology and advanced products to providing the highest level of customer service and satisfaction in the industry. Producing products to exacting quality standards that provide excellent, long-term reliability is given the same level of importance and priority as device performance. IDT is also dedicated to delivering these high-quality advanced products on time. The company would like to be known not only for its technological capabilities, but also for providing its customers with quick, responsive and courteous service.

IDT's product families are available in both commercial and military grades. As a bonus, commercial customers obtain the benefits of military processing disciplines, established to meet or exceed the stringent criteria of the applicable military specifications.

IDT is the leading U.S. supplier of high-speed CMOS circuits. The company's high-performance fast SRAM , FCT logic family, high-density modules, FIFOs, complex logic products, specialty memories, ECL I/O BiCEMOS™ memories, RISC subsystems, and the 32-bit RISC microprocessor family complement each other to provide high-speed CMOS solutions to a wide range of applications and systems.

Dedicated to maintaining its leadership position as a state-of-the-art IC manufacturer, IDT will continue to focus on maintaining its technology edge as well as developing a broader range of innovative products. New products and speed enhancements are continuously being added to each of the existing product families and additional product lines will be introduced. Contact your IDT field representative or factory marketing engineer to determine the latest product offerings. If you're building state-of-the-art equipment, IDT wants to help you solve some of your design problems.

IDT MILITARY AND DESC-SMD PROGRAM

IDT is a leading supplier of military, high-speed CMOS circuits. The company's high-performance Static RAMs, FCT Logic Family, Complex Logic (CLP), FIFOs, Specialty Memories (SMP), ECL I/O BiCMOS Memories, 32-bit RISC Microprocessor, RISC Subsystems and high-density Subsystems Modules product lines complement each other to provide high-speed CMOS solutions to a wide range of military applications and systems. Most of these product lines offer Class B products which are fully compliant to the latest revision of MIL-STD-883, Paragraph 1.2.1. In addition, IDT offers Radiation Tolerant (RT), as well as Radiation Enhanced (RE), products.

IDT has an active program with the Defense Electronic Supply Center (DESC) to list all of IDT's military compliant devices on Standard Military Drawings (SMD). The SMD program allows standardization of militarized products and reduction of the proliferation of non-standard source control drawings. This program will go far toward reducing the need for each defense contractor to make separate specification control drawings for purchased parts. IDT plans to have SMDs for many of its product offerings. Presently, IDT has 88 devices which are listed or pending listing. The devices are from IDT's SRAM, FCT Logic family, Complex Logic (CLP), FIFOs and Specialty Memories (SMP) product families. IDT expects to add another 20 devices to the SMD program in the near future. Users should contact either IDT or DESC for current status of products in the SMD program.

SMD	
SRAM	**IDT**
84036/D	6116
5962-88740	6116LA
84132/B	6167
5962-86015/A	7187
5962-86859	6198/7198/7188
5962-86705/A	6168
5962-85525/A	7164
5962-88552	71256L
5962-88662	71256S
5962-88611	71682L
5962-88681/A	71258S
5962-88545	71258L
5962-88544	71257L
5962-88725/A	71257S
5962-89690	6116
5962-89691	7198
5962-89692	7188
5962-89712	71982
5962-89892	6198
5962-38294	8Kx8
5962-89598	32Kx8
SMP	**IDT**
5962-86875/A	7130/7140
5962-87002/A	7132/7142
5962-88610/A	7133S/7143S
5962-88665/A	7133L/7143L
CLP	**IDT**
5962-87708/A	39C10B & C
5962-88535	39C01
5962-88533/A	49C460A
5962-88613	39C60A
5962-88643	49C410
5962-88743	75C48S
5962-86273	7216L
5962-87686	7217L
5962-88733	7210
5962-90579	7381L

SMD	
LOGIC	**IDT**
5962-87630/B	54FCT244/A
5962-87629/C	54FCT245/A
5962-86862/A	54FCT299/A
5962-87644/A	54FCT373/A
5962-87628/C	54FCT374/A
5962-87627	54FCT377/A
5962-87654/A	54FCT138/A
5962-87655	54FCT240/A
5962-87656/A	54FCT273/A
5962-89533	54FCT861A/B
5962-89506	54FCT827A/B
5962-88575	54FCT841A/B
5962-88608	54FCT821A/B
5962-88543/A	54FCT521/A
5962-88640	54FCT161/A
5962-88639	54FCT573/A
5962-88656	54FCT823A/B
5962-88657	54FCT163/A
5962-88674	54FCT825A/B
5962-88661	54FCT863A/B
5962-88736	54FCT520A/B
5962-88775	54FCT646A/B
5962-89508	54FCT139/A
5962-89665	54FCT824A/B
5962-88651	54FCT533/A
5962-88652	54FCT182/A
5962-88653	54FCT645A/B
5962-88654	54FCT640A/B
5962-88655	54FCT534/A
5962-89767	54FCT540/A
5962-89766	54FCT541/A
5962-89733	54FCT191/A
5962-89732	54FCT241/A
5962-89652	54FCT399/A
5962-89513	54FCT574/A
5962-89731	54FCT833A/B
5962-88675	54FCT845A/B
5962-89730	54FCT543/A

SMD	
FIFO	**IDT**
5962-87531	7201LA
5962-86846/A	72404
5962-88669	7203S
5962-89568	7204L
5962-89536	7202L
5962-89863	7201S
5962-89523	72403L
5962-89666	7200L
5962-89942	72103L
5962-89943	72104L
5962-89567	7203L

2509 tbl 01

RADIATION HARDENED TECHNOLOGY

IDT manufactures and supplies radiation hardened products for military/aerospace applications. Utilizing special processing and starting materials, IDT's radiation hardened devices are able to survive in hostile radiation environments. In total dose, dose rate and environments where single event upset is of concern, IDT products are designed to continue functioning without loss of performance. IDT can supply all its products on these processes. Total Dose radiation testing is performed in-house on an ARACOR X-Ray system. External facilities are utilized for device research on gamma cell, LINAC and other radiation equipment. IDT has an on-going research and development program for improving radiation handling capabilities (See "IDT Radiation Tolerant/Enhanced Products for Radiation Environments" in Section 3) of IDT products/processes.

IDT LEADING EDGE CEMOS TECHNOLOGY

HIGH-PERFORMANCE CEMOS

From IDT's beginnings in 1980, it has had a belief in and a commitment to CMOS. The company developed a high-performance version of CMOS, called enhanced CMOS (CEMOS), that allows the design and manufacture of leading-edge components. It incorporates the best characteristics of traditional CMOS, including low power, high noise immunity and wide operating temperature range; it also achieves speed and output drive equal or superior to bipolar Schottky TTL. The last decade has seen development and production of four "generations" of IDT's CEMOS technology with process improvements which have reduced IDT's electrical effective (Leff) gate lengths by more than 50 percent from 1.3 microns (millionths of a meter) in 1981 to 0.6 microns in 1989.

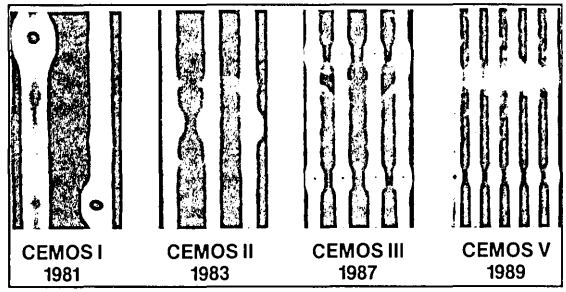
	CEMOS I	CEMOS II A	CEMOS II C	CEMOS III	CEMOS V	CEMOS VI
Calendar Year	**1981**	**1983**	**1985**	**1987**	**1989**	**1990**
Drawn Feature Size	2.5μ	1.7μ	1.3μ	1.2μ	1.0μ	0.8μ
Leff	1.3μ	1.1μ	0.9μ	0.8μ	0.6μ	0.45μ
Basic Process Enhancements	Dual-well, Wet Etch, Projection Aligned	Dry Etch, Stepper	Shrink, Spacer	Silicide, BPSG, BiCEMOS I	BiCEMOS II	BiCEMOS III

2514 drw 01

CEMOS IV = CEMOS III – scaled process optimized for high-speed logic.

Figure 1.

Continual advancement of CEMOS technology allows IDT to implement progressively higher levels of integration and achieve increasingly faster speeds maintaining the company's established position as the leader in high-speed CMOS integrated circuits. In addition, the fundamental process technology has been extended to add bipolar elements to the CEMOS platform. IDT's BiCEMOS process combines the ultra-high speeds of bipolar devices with the lower power and cost of CMOS, allowing us to build even faster components than straight CMOS at a slightly higher cost.



SEM photos (miniaturization)

2514 drw 02

Figure 2. Fifteen-Hundred-Power Magnification Scanning Electron Microscope (SEM) Photos of the Four Generations of IDT's CEMOS Technology

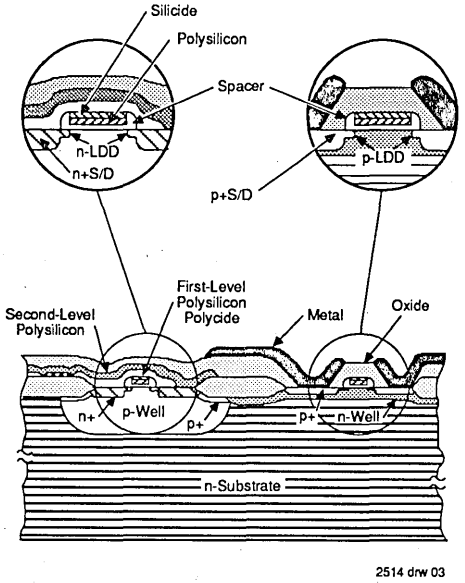


Figure 3. IDT CEMOS Device Cross Section

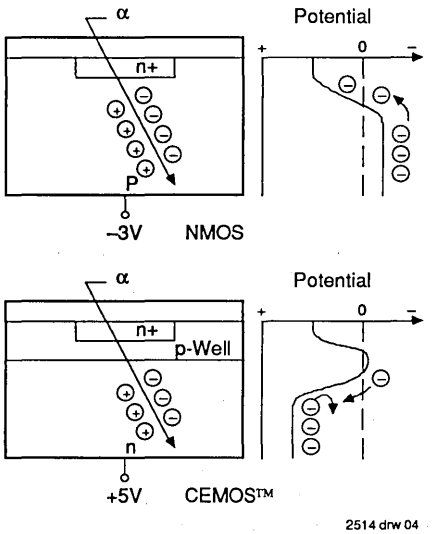


Figure 4. IDT CEMOS Built-In High Alpha Particle Immunity

ALPHA PARTICLES

Random alpha particles can cause memory cells to temporarily lose their contents or suffer a "soft error." Traveling with high energy levels, alpha particles penetrate deep into an integrated chip. As they burrow into the silicon, they leave a trail of free electron-hole pairs in their wake.

The cause of alpha particles is well documented and understood in the industry. IDT has considered various techniques to protect the cells from this hazardous occurrence. These techniques include dual-well structures (Figures 3 and 4) and a polymeric compound for die coating. Presently, a polymeric compound is used in many of IDT's SRAMs; however, the specific techniques used may vary and change from one device generation to the next as the industry and IDT improve the alpha particle protection technology.

LATCHUP IMMUNITY

A combination of careful design layout, selective use of guard rings and proprietary techniques have resulted in virtual elimination of latchup problems often associated with older CMOS processes (Figure 5). The use of NPN and N-channel I/O devices eliminates hole injection latchup. Double guard ring structures are utilized on all input and output circuits to absorb injected electrons. These effectively cut off the current paths into the internal circuits to essentially isolate I/O circuits. Compared to older CMOS processes which exhibit latchup characteristics with trigger currents from 10-20mA, IDT products inhibit latchup at trigger currents substantially greater than this.

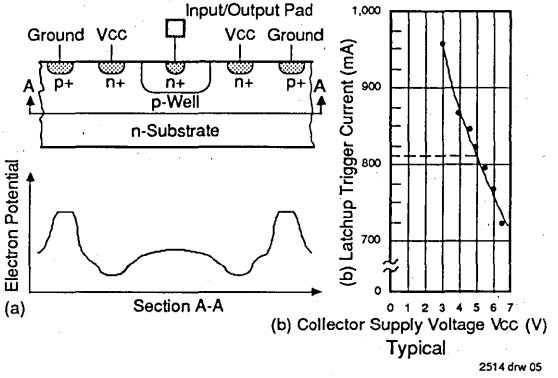


Figure 5. IDT CEMOS Latchup Suppression

SURFACE MOUNT TECHNOLOGY AND IDT'S MODULE PRODUCTS

Requirements for circuit area reduction, utilizing the most efficient and compact component placement possible and the needs of production manufacturing for electronics assemblies are the driving forces behind the advancement of circuit-board assembly technologies. These needs are closely associated with the advances being made in surface mount devices (SMD) and surface mount technology (SMT) itself. Yet, there are two major issues with SMT in production manufacturing of electronic assemblies: high capital expenditures and complexity of testing.

The capital expenditure required to convert to efficient production using SMT is still too high for the majority of electronics companies, regardless of the 20-60% increase in the board densities which SMT can bring. Because of this high barrier to entry, we will continue to see a large market segment [large even compared to the exploding SMT market] using traditional through-hole packages (i.e. DIPs, PGAs, etc) and assembly techniques. How can these types of companies take advantage of SMD and SMT? Let someone else, such as IDT, do it for them by investing time and money in SMT and then in return offer through-hole products utilizing SMT processes. Products which fit this description are multi-chip modules, consisting of SMT assembled SMDs on a through-hole type substrate. Modules enable companies to enjoy SMT density advantages and traditional package options without the expensive startup costs required to do SMT in-house.

Although subcontracting this type of work to an assembly house is an alternative, there still is the other issue of testing, an area where many contract assembly operations fall short of IDT's capability and experience. Prerequisites for adequate module testing sophisticated high performance parametric testers, customized test fixtures, and most importantly the experience to tests today's complex electronic devices. Companies can therefore take advantage of IDT's experience in testing and manufacturing high performance CMOS multi-chip modules.

At IDT, SMD components are electrically tested, environmentally screened, and performance selected for each IDT module. All modules are 100% tested as if they are a separate functional component and are guaranteed to meet all specified parameters at the module output without the customer having to understand the modules' internal workings.

Other added benefits companies get by using IDT's CMOS module products are:

1) a wide variety of high performance, through-hole products utilizing SMD packaged components,
2) fast speeds compared with NMOS based products,
3) low power consumption compared with bipolar technologies, and
4) low cost manufacturability compared with GaAs based products.

IDT has recognized the problems of SMT and began offering CMOS modules as part of its standard product portfolio. IDT modules combine the advantages of:

1) the low power characteristics of IDT's CEMOS™ and BiCEMOS™ products,
2) the density advantages of first class SMD components including those from IDT's components divisions, and
3) experience in system level design, manufacturing, and testing with its own in-house SMT operation.

IDT currently has two divisions (Subsystems and RISC Subsystems) dedicated to the development of module products ranging from simple memory modules to complex VME sized application specific modules to full system level CPU boards. These modules have surface mount devices assembled on both sides of either a multi-layer glass filled epoxy (FR-4) or a multi-layer co-fired ceramic substrate. Assembled modules come available in industry standard through-hole packages and other space-saving module packages. Industry proven vapor-phase or IR reflow techniques are used to solder the SMDs to the substrate during the assembly process. Because of our affiliation with IDT's experienced semiconductor manufacturing divisions, we thoroughly understand and therefore test all modules to the applicable datasheet specifications and customer requirements.

Thus, IDT is able to offer today's electronic design engineers a unique solution for their "need-more-for-less" problem.modules. These high speed, high performance products offer the density advantages of SMD and SMT, the added benefit of low power CMOS technology, and through-hole packaged electronics without the high cost of doing it in-house.

STATE-OF-THE-ART FACILITIES AND CAPABILITIES

Integrated Device Technology is headquartered in Santa Clara, California — the heart of the "Silicon Valley." The company's operations are housed in seven facilities totaling over 500,000 square feet. These facilities house all aspects of business from research and development to design, wafer fabrication, assembly, environmental screening, test and administration. In-house capabilities include scanning electron microscope (SEM) evaluation, particle impact noise detection (PIND), plastic and hermetic packaging, military and commercial testing, burn-in, life test and a full complement of environmental screening equipment.

The over-200,000-square-foot corporate headquarters campus is composed of four buildings. The largest facility on this site is a 100,000 square foot, two-building complex. The first building, a 60,000 square foot facility, is dedicated to the Complex Logic, Standard Logic and RISC Microprocessor product lines, as well as hermetic and plastic package assembly, logic products' test, burn-in, mark and QA, and a reliability/failure analysis lab.

IDT's Packaging and Assembly Process Development teams are located here. To keep pace with the development of new products and to enhance the IDT philosophy of "Innovation," these teams have ultra modern, integrated and correspondingly sophisticated equipment and environments at their disposal. All manufacturing is completed in dedicated clean room areas (Class 10K minimum), with all preseal operations accomplished under Class 100 laminar flow hoods.

Development of assembly materials, processes and equipment is accomplished under a fully operational production environment to ensure reliability and repeatable product. The Hermetic Manufacturing and Process Development team is currently producing custom products to the strict requirements of MIL-STD-883. The fully automated plastic facility is currently producing high volumes of USA-manufactured product, while developing state-of-the-art surface mount technology patterned after MIL-STD-883.

The second building of the complex houses sales, marketing, finance and MIS.

The RISC Subsystems and Subsystems Modules Divisions are located behind the two-building complex in a 54,000 square foot facility. Also located at this facility are Quality Assurance and wafer fabrication services.

Directly across the street from the two-building complex is a newly acquired 50,000 square foot facility that houses administrative services, Northwest Area Sales, Human Resources, International Planning and Shipping and Receiving functions.

IDT's largest and newest facility, opened in 1990 in San Jose, California, is a multi-purpose 150,000 square foot, ultra modern technology development center. This facility houses a 25,000 square foot, combined Class 1 (a maximum of one particle per cubic foot of 0.2 micron or larger), sub-half-micron R&D fabrication facility and a wafer fabrication area. This fab supports both production volumes of IDT products, including some next generation SRAMs, and the R&D efforts of the technology development staff. Technology development efforts targeted for the center include advanced silicon processing and wafer fabrication techniques. A test area to support both production and research is located on-site. The building is also the new home of the FIFO and ECL product lines.

IDT's second largest facility is located in Salinas, California, about an hour away from Santa Clara. This 95,000 square foot facility, located on 14 acres, is the Static RAM Division and Specialty Memory product line. Constructed in 1985, this facility houses an ultra-modern 25,000 square foot high-volume wafer fabrication area measured at Class 2-to-3 (a maximum of 2 to 3 particles per cubic foot of 0.2 micron or larger) clean room conditions. Careful design and construction of this fabrication area created a clean room environment far beyond the 1985 average for U.S. fab areas. This made possible the production of large volumes of high-density submicron geometry, fast static RAMs. This facility also houses shipping areas for IDT's leadership family of CMOS static RAMs. This site will expand to accommodate a 250,000 square foot complex.

To extend these philosophies while maintaining strict control of our processes, IDT has an operational Assembly and Test facility located in Penang, Malaysia. This facility assembles product to USA standards, with all assemblies done under laminar flow conditions (Class 100) until the silicon is encased in its final packaging. All products in this facility are manufactured to the quality control requirements of MIL-STD-883.

All of IDT's facilities are aimed at increasing our manufacturing productivity to supply ever larger volumes of high-performance, cost-effective leadership CMOS products.

SUPERIOR QUALITY AND RELIABILITY

Maintaining the highest standards of quality in the industry on all products is the basis of Integrated Device Technology's manufacturing systems and procedures. From inception, quality and reliability are built into all of IDT's products. Quality is "designed in" at every stage of manufacturing – as opposed to being "tested-in" later – in order to ensure impeccable performance.

Dedicated commitment to fine workmanship, along with development of rigid controls throughout wafer fab, device assembly and electrical test, create inherently reliable products. Incoming materials are subjected to careful inspections. Quality monitors, or inspections, are performed throughout the manufacturing flow.

IDT military grade monolithic hermetic products are designed to meet or exceed the demanding Class B reliability levels of MIL-STD-883 and MIL-M-38510, as defined by Paragraph 1.2.1 of MIL-STD-883.

Product flow and test procedures for all monolithic hermetic military grade products are in accordance with the latest revision and notice of MIL-STD-883. State-of-the-art production techniques and computer-based test procedures are coupled with tight controls and inspections to ensure that products meet the requirements for 100% screening. Routine quality conformance lot testing is performed as defined in MIL-STD-883, Methods 5004 and 5005.

For IDT module products, screening of the fully assembled substrates is performed, in addition to the monolithic level screening, to assure package integrity and mechanical reliability. All modules receive 100% electrical tests (DC, functional and dynamic switching) to ensure compliance with the "subsystem" specifications.

By maintaining these high standards and rigid controls throughout every step of the manufacturing process, IDT ensures that commercial, industrial and military grade products consistently meet customer requirements for quality, reliability and performance.

SPECIAL PROGRAMS

Class S. IDT also has all manufacturing, screening and test capabilities in-house (except X-ray and some Group D tests) to perform complete Class S processing per MIL-STD-883 on all IDT products and has supplied Class S products on several programs.

Radiation Hardened. IDT has developed and supplied several levels of radiation hardened products for military/aerospace applications to perform at various levels of dose rate, total dose, single event upset (SEU), upset and latchup. IDT products maintain nearly their same high-performance levels built to these special process requirements. The company has in-house radiation testing capability used both in process development and testing of deliverable product. IDT also has a separate group within the company dedicated to supplying products for radiation hardened applications and to continue research and development of process and products to further improve radiation hardening capabilities.

QSP–QUALITY, SERVICE AND PERFORMANCE

Quality from the beginning, is the foundation for IDT's commitment to supply consistently high-quality products to our customers. IDT's quality commitment is embodied in its all pervasive Constant Quality Improvement (CQI) program. Everyone who influences the quality of the product–from the designer to the shipping clerk–is committed to constantly improving the product quality.

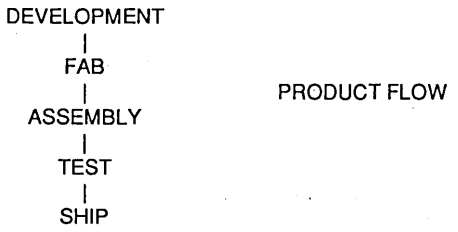
IDT'S FOCUS

"To make quantitative constant improvement in the quality of our actions that result in the supply of leadership products in conformance to the requirements of our customers."

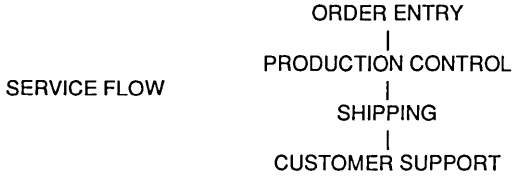
IDT has dedicated its efforts to constant quantitative improvements in quality. The result, a supply of leadership products that conform to the requirements of our customers.

IDT'S PRODUCT ASSURANCE STRATEGY FOR CQI

Measurable standards are essential to the success of CQI. All the processes contributing to the final quality of the product need to be monitored, measured and improved upon through the use of statistical tools.



Our customers receive the benefit of our optimized systems. Installed to enhance quality and reliability, these systems provide accurate and timely reporting on the effectiveness of manufacturing controls and the reliability and quality performance of IDT products and services.



These systems and controls concentrate on CQI by focusing on the following key elements:

Statistical Techniques
Using statistical techniques, including Statistical Process Control (SPC) to determine whether the product/ processes are under control.

Standardization
Implementing policies, procedures and measurement techniques that are common across different operational areas.

Documentation
Documenting and training in policies, procedures, measurement techniques and updating through characterization/ capability studies.

Productivity Improvement
Using constant improvement teams made up from employees at all levels of the organization.

Leadership
Focusing on quality as a key business parameter and strategic strength.

Total Employee Participation
Incorporating the CQI program into the IDT Corporate Culture.

Customer Service
Supporting the customer, as a partner, through performance review and pro-active problem solving.

People Excellence
Committing to growing, motivating and retaining people through training, goal setting, performance measurement and review.

PRODUCT FLOW

Product quality starts here. IDT has mechanisms and procedures in place that monitor and control the quality of our development activities. From the calibration of design capture libraries through process technology and product characterization that establish whether the performance, ratings and reliability criteria have been met. This includes failure analysis of parts that will improve the prototype product.

At the pre-production stage once again in-house qualification tests assure the quality and reliability of the product. All specifications and manufacturing flows are established and personnel trained before the product is placed into production.

Manufacturing

To make CQI during the manufacturing stage, control items are determined for major manufacturing conditions. Data is gathered and statistical techniques are used to control specific manufacturing processes that affect the quality of the product.

In-process and final inspections are fed back to earlier processes to improve product quality. All product is burned-in (where applicable) before 100% inspection of electrical characteristics takes place.

Products which pass final inspection are then subject to Quality Assurance and Reliability Tests. This data is used to improve manufacturing processes and provide reliability predictions of field applications.

Inventory and Shipping

Controls in shipping focus on ensuring parts are identified and packaged correctly. Care is also taken to see that the correct paperwork is present and the product being shipped was processed correctly.

SERVICE FLOW

Quality not only applies to the product but to the quality -of -service we give our customers. Service is also constantly improved.

Order Procedures

Checks are made at the order entry stage to ensure the correct processing of the Customer's product. After verification and data entry the Acknowledgements (sent to Customers) are again checked to ensure details are correct. As part of the CQI program, the results of these verifications are analyzed using statistical techniques and corrective actions are taken.

Production Control

Production Control (P.C.) is responsible for the flow and logistics of material as it moves through the manufacturing processes. The quality of the actions taken by P.C. greatly impinges on the quality of service the customer receives. Because many of our customers have implemented Just-in-Time (JIT) manufacturing practices, IDT as a supplier also has to adopt these same disciplines. As a result, employees receive extensive training and the performance level of key actions are kept under constant review. These key actions include:

Quotation response and accuracy.
Scheduling response and accuracy.
Response and accuracy of Expedites.
Inventory, management, and effectiveness.
On time delivery.

Customer Support

IDT has a worldwide network of sales offices and Technical Development Centers. These provide local customer support on business transactions, and in addition, support customers on applications information, technical services, benchmarking of hardware solutions, and demonstration of various Development Workstations.

The key to CQI is the timely resolution of defects and implementation of the corrective actions. This is no more important than when product failures are found by a customer. When failures are found at the customer's incoming inspection, in the production line, or the field application, the Quality Assurance group is the focal point for the investigation of the cause of failure and implementation of the corrective action. IDT constantly improves the level of support we give our customers by monitoring the response time to customers that have detected a product failure. Providing the customer with an analysis of the failure, including corrective actions and the statistical analysis of defects, brings CQI full circle–full support of our customers and their designs with high-quality products.

SUMMARY

In 1990, IDT made the commitment to *"Leadership through Quality, Service, and Performance Products"*.

We believe by following that credo IDT and our cusotmers will be successful in the coming decade. With the implementation of the CQI strategy, we will satisfy our goal...

"Leadership through Quality, Service and Performance Products".

IDT QUALITY CONFORMANCE PROGRAM

A COMMITMENT TO QUALITY

Integrated Device Technology's monolithic and modular assembly products are designed, manufactured and tested in accordance with the strict controls and procedures required by Military Standards. The documentation, design and manufacturing criteria of the Quality and Reliability Assurance Program were developed and are being maintained to the most current revisions of MIL-38510 as defined by paragraph 1.2.1 of MIL-STD-883 and MIL-STD-883 requirements.

Product flow and test procedures for all Class B *monolithic* hermetic Military Grade microcircuits are in full compliance with paragraph 1.2.1 of MIL-STD-883. State-of-the-art production techniques and computer-based test procedures are coupled with stringent controls and inspections to ensure that products meet the requirements for 100% screening and quality conformance tests as defined in MIL-STD-883, Methods 5004 and 5005.

Product flow and test procedures for all *plastic* and *commercial hermetic* products are in accordance with industry practices for producing highly reliable microcircuits to ensure that products meet the IDT requirements for 100% screening and quality conformance tests.

By maintaining these high standards and rigid controls throughout every step of the manufacturing process, IDT ensures that our products consistently meet customer requirements for quality, reliability and performance.

SUMMARY

Monolithic Hermetic Package Processing Flow[1]

Refer to the Monolithic Hermetic Package Processing Flow diagram. All test methods refer to MIL-STD-883 unless otherwise stated.

1. **Wafer Fabrication:** Humidity, temperature and particulate contamination levels are controlled and maintained according to criteria patterned after Federal Standard 209, Clean Room and Workstation Requirements. All critical workstations are maintained at Class 100 levels or better.

   Wafers from each wafer fabrication area are subjected to Scanning Electron Microscope analysis on a periodic basis.

2. **Die-Sort Visual Inspection:** Wafers are cut and separated and the individual die are 100% visually inspected to strict IDT-defined internal criteria.

3. **Die Shear Monitor:** To ensure die attach integrity, product samples are routinely subjected to a shear strength test per Method 2019.

4. **Wire Bond Monitor:** Product samples are routinely subjected to a strength test per Method 2011, Condition D, to ensure the integrity of the lead bond process.

5. **Pre-Cap Visual:** Before the completed package is sealed, 100% of the product is visually inspected to Method 2010, Condition B criteria.

6. **Environmental Conditioning:** 100% of the sealed product is subjected to environmental stress tests. These thermal and mechanical tests are designed to eliminate units with marginal seal, die attach or lead bond integrity.

7. **Hermetic Testing:** 100% of the hermetic packages are subjected to fine and gross leak seal tests to eliminate marginally sealed units or units whose seals may have become defective as a result of environmental conditioning tests.

8. **Pre-Burn-In Electrical Test:** Each product is 100% electrically tested at an ambient temperature of +25°C to IDT data sheet or the customer specification.

9. **Burn-In:** 100% of the Military Grade product is burned-in under dynamic electrical conditions to the time and temperature requirements of Method 1015, Condition D. Except for the time, Commercial Grade product is burned-in as applicable to the same conditions as Military Grade devices.

10. **Post-Burn-In Electrical:** After burn-in, 100% of the Class B Military Grade product is electrically tested to IDT data sheet or customer specifications over the –55°C to +125°C temperature range. Commercial Grade products are sample tested to the applicable temperature extremes.

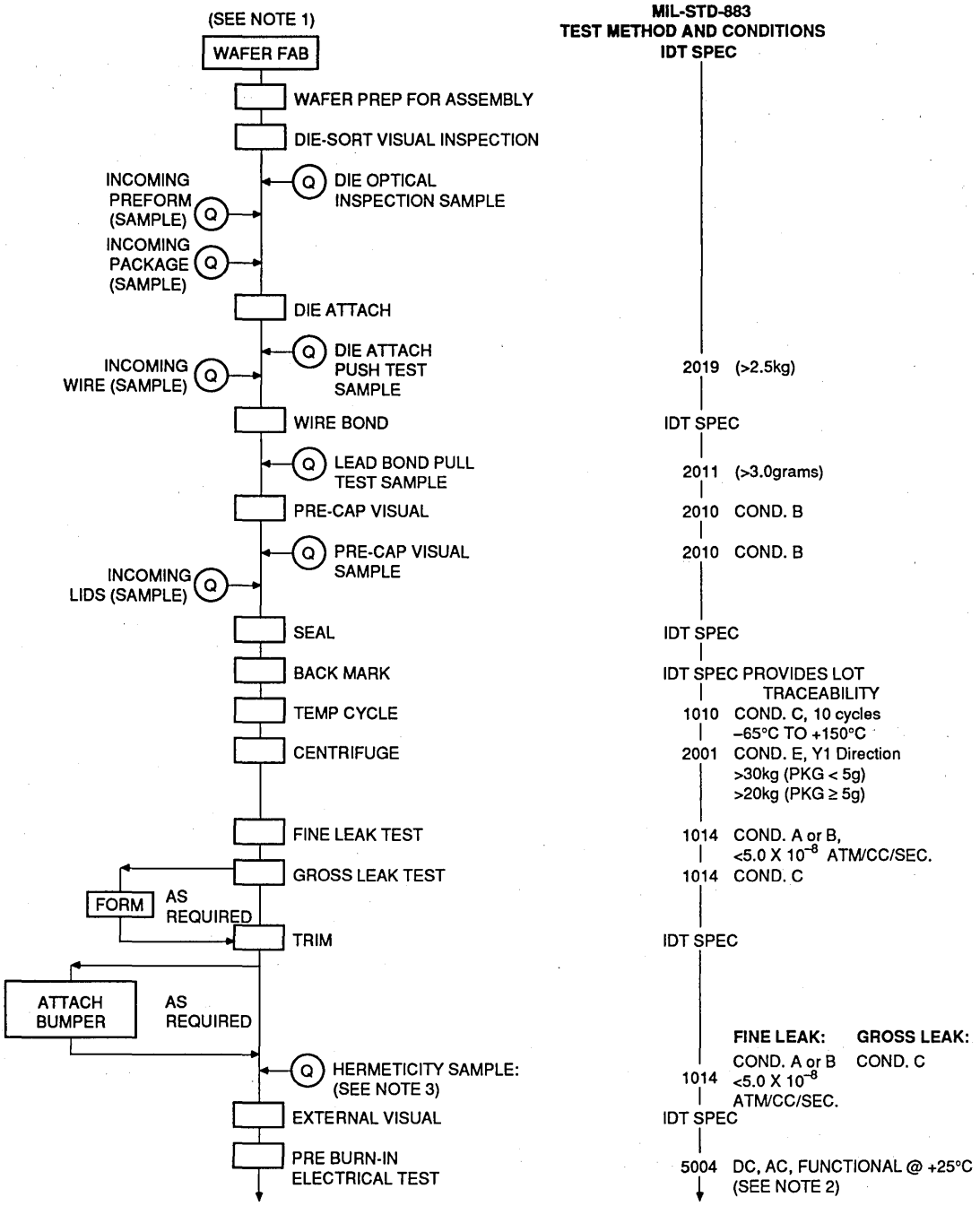
11. **Mark:** All product is marked with product type and lot code identifiers. MIL-STD-883 compliant Military Grade products are identified with the required compliant code letter.

12. **Quality Conformance Tests:** Samples of the Military Grade product which have been processed to the 100% screening tests of Method 5004 are routinely subjected to the quality conformance requirements of Method 5005.

NOTE:

1. For quality requirements beyond Class B levels such as SEM analysis, X-Ray inspection, Particle Impact Noise Reduction (PIND) test, Class S screening or other customer specified screening flows, please contact your Integrated Device Technology sales representative.

Monolithic Hermetic Package Processing Flow



SEE FINAL PROCESSING FLOW FOR REMAINDER OF OPERATIONS AND NOTES

2502 drw 01

SUMMARY

Monolithic Plastic Package Processing Flow

Refer to the Monolithic Plastic Package Processing Flow diagram. All test methods refer to MIL-STD-883 unless otherwise stated.

1. **Wafer Fabrication:** Humidity, temperature and particulate contamination levels are controlled and maintained according to criteria patterned after Federal Standard 209, Clean Room and Workstation Requirements. All critical workstations are maintained at Class 100 levels or better.

   Topside silicon nitride passivation is all applied to all wafers for better moisture barrier characteristics.

   Wafers from each wafer fabrication area are subjected to Scanning Electron Microscope analysis on a periodic basis.

2. **Die-Sort Visual Inspection:** Wafers are 100% visually inspected to strict IDT defined internal criteria.

3. **Die Push Test:** To ensure die attach integrity, product samples are routinely subjected to die push tests.

4. **Wire Bond Monitor:** Product samples are routinely subjected to wire bond pull tests to ensure the integrity of the lead bond process.

5. **Pre-Cap Visual:** Before the package is molded, 100% of the product is visually inspected to criteria patterned after MIL-STD-883, Method 2010, Condition B.

6. **Post Mold Cure:** Plastic encapsulated devices are baked to ensure an optimum plastic seal so as to enhance moisture barrier characteristics.

7. **Pre-Burn-In Electrical:** Each product is 100% electrically tested at an ambient temperature of +25°C to IDT data sheet or the customer specification.

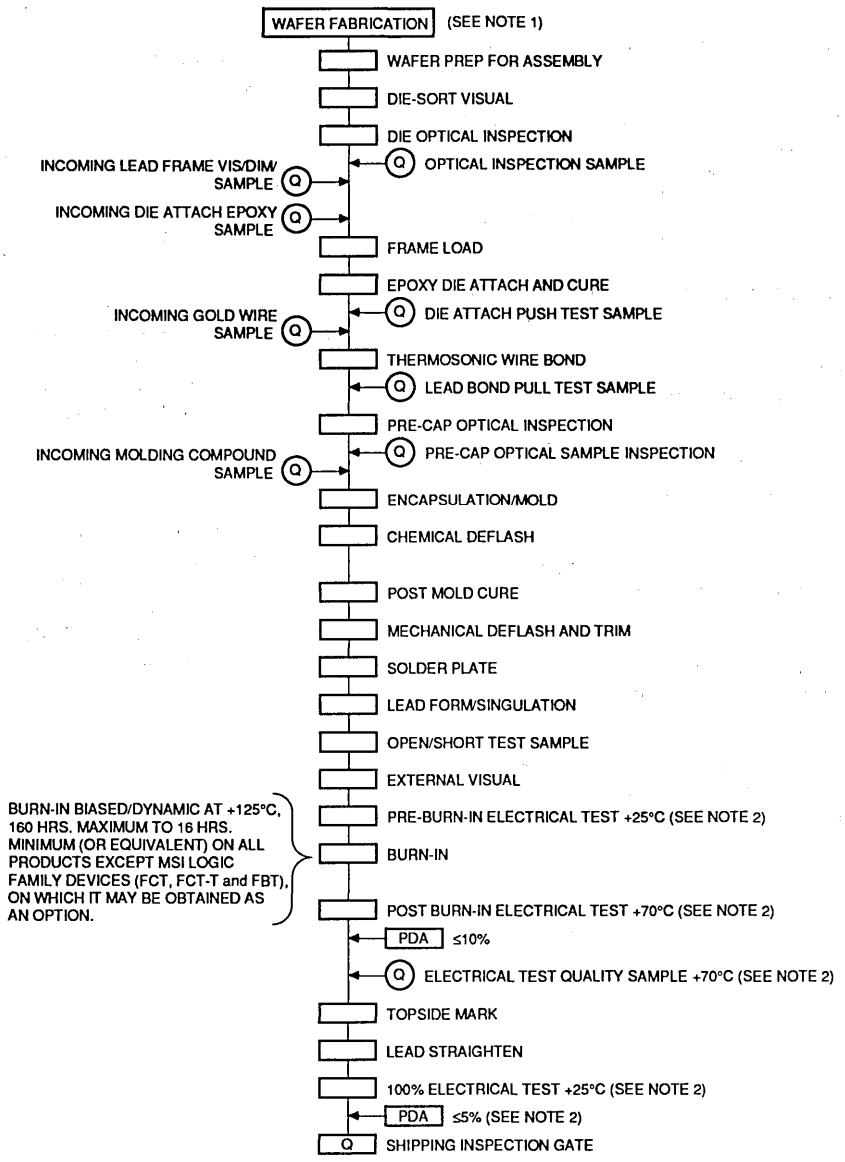
8. **Burn-In:** Except for MSI Logic family devices where it may be obtained as an option, all Commercial Grade plastic package products are burned-in 16 hours at +125°C (or equivalent), utilizing the same burn-in conditions as the Military Grade product.

9. **Post-Burn-In Electrical:** After burn-in, 100% of the plastic product is electrically tested to IDT data sheet or customer specifications at the maximum temperature extreme. The minimum temperature extreme is tested periodically on an audit basis.

10. **Mark:** All product is marked with product type and lot code identifiers.

11. **Quality Conformance Inspection:** Samples of the plastic product which have been processed to the 100% screening requirements are subjected to the Periodic Quality Conformance Inspection Program. Where indicated, the test methods are patterned after MIL-STD-883 criteria.

Monolithic Plastic Package Processing Flow



NOTE:
1. All screens are 100% unless otherwise noted.
2. All electrical test programs are per the applicable IDT test specification.
3. IDT performs a 100% electrical test at +25°C with a 5% PDA limit at this point.
4. (Q) = Quality sample inspection.

2502 drw 02

Monolithic Hermetic Package Final Processing Flow

Operation	MIL-STD-883 Test Method	Military Compliant Class B	Commercial	
			Military Temp. Range	Commercial Temp. Range
Burn-In	1015/D at +125°C Min. or Equivalent	100% 160 Hours	100% 16 to 160 Hours	100% 16to160 Hours
Post Burn-In Electrical: Static (DC), Functional and Switching (AC)[2]	IDT Spec.	100% +25, -55 and +125°C	100% +125°C	100% +70°C
Percent Defective Allowed (PDA)[4]	5004 or IDT Spec.	5%	10%	10%
Group A Electrical: Static (DC), Functional and Switching (AC)[2]	5005 and IDT Spec.	Sample -55 and +125°C	Sample +125°C	Sample +70°C
Mark/Lead Straighten	IDT Spec.	100%	100%	100%
+25°C Electrical[2]	IDT Spec.	100%[5]	100%	100%
Final Visual/Pack	IDT Spec.	100%	100%	100%
Quality Conformance Inspection	5005 (Group B, C, D)	Yes	—	—
Quality Shipping Inspection (Visual/Plant Clearance)	IDT Spec.	Sample	Sample	Sample

2505 tbl 01

NOTES:
1. All screens are 100% unless otherwise noted.
2. All electrical test programs are per the applicable IDT test specification.
3. This hermeticity sample is performed after all lead finish operations.
4. If a lot fails the 5% PDA but is ≤10%, the lot may be resubmitted to burn-in one time only to the same time and temperature conditions as first submission. The subsequent post burn-in electrical test at +25°C will be performed to a PDA of 3%.
5. IDT performs a 100% electrical test at +25°C with a 2% PDA limit at this point to satisfy group A requirements, and considers this to be equivalent to the group A requirement of an LTPD of 2, with an accept number of 0. If a lot fails the 2% PDA limit, it may be rescreened one time only to a tightened PDA limit of 1.5%.
6. Ⓠ = Quality sample inspection.

RADIATION TOLERANT/ENHANCED/HARDENED PRODUCTS FOR RADIATION ENVIRONMENTS

INTRODUCTION

The need for high-performance CMOS integrated circuits in military and space systems is more critical today than ever before. The low power dissipation that is achieved using CMOS technology, along with the high complexity and density levels, makes CMOS the nearly ideal component for all types of applications.

Systems designed for military or space applications are intended for environments where high levels of radiation may be encountered. The implication of a device failure within a military or space system clearly is critical. IDT has made a significant contribution toward providing reliable radiation-tolerant systems by offering integrated circuits with enhanced radiation tolerance. Radiation environments, IDT process enhancements and device tolerance levels achieved are described below.

THE RADIATION ENVIRONMENT

There are four different types of radiation environments that are of concern to builders of military and space systems. These environments and their effects on the device operation, summarized in Figure 1, are as follows:

Total Dose Accumulation refers to the total amount of accumulated gamma rays experienced by the devices in the system, and is measured in RADS (SI) for radiation units experienced at the silicon level. The physical effect of gamma rays on semiconductor devices is to cause threshold shifts (Vt shifts) of both the active transistors as well as the parasitic field transistors. Threshold voltages decrease as total dose is accumulated; at some point, the device will begin to exhibit parametric failures as the input/output and supply currents increase. At higher radiation accumulation levels, functional failures occur. In memory circuits, however, functional failures due to memory cell failure often occur first.

Burst Radiation or Dose Rate refers to the amount of radiation, usually photons or electrons, experienced by the devices in the system due to a pulse event, and is measured in RADS (SI) per second. The effect of a high dose rate or burst of radiation on CMOS integrated circuits is to cause temporary upset of logic states and/or CMOS latch-up. Latch-up can cause permanent damage to the device.

Single Event Upset (SEU) is a transient logic state change caused by high-energy ions, such as energetic cosmic rays, striking the integrated circuits. As the ion passes through the silicon, charge is either created through ionization or direct nuclear collision. If collected by a circuit node, this excess charge can cause a change in logic state of the circuit. Dynamic nodes that are not actively held at a particular logic state (dynamic RAM cells for example) are the most susceptible. These upsets are transient, but can cause system failures known as "soft errors."

Neutron Irradiation will cause structural damage to the silicon lattice which may lead to device leakage and, ultimately, functional failure.

Radiation Category	Primary Particle	Source	Effect
Total Dose	Gamma	Space or Nuclear Event	Permanent
Dose Rate	Photons	Nuclear Event	Temporary Upset of Logic State or Latch-up
SEU	Cosmic Rays	Space	Temporary Upset of Logic State
Neutron	Neutrons	Nuclear Event	Device Leakage Due to Silicon Lattice Damage

2510 drw 01

Figure 1.

DEVICE ENHANCEMENTS

Of the four radiation environments above, IDT has taken considerable data on the first two, Total Dose Accumulation and Dose Rate. IDT has developed a process that significantly improves the radiation tolerance of its devices within these environments. Prevention of SEU failures is usually accomplished by system-level considerations, such as Error Detection and Correction (EDC) circuitry, since the occurrence of SEUs is not particularly dependent on process technology. Through IDT's customer contracts, SEU has been gathered on some devices. Little is yet known about the effects of neutron-induced damage. For more information on SEU testing, contact IDT's Radiation Hardened Product Group.

Enhancements to IDT's standard process are used to create radiation enhanced and tolerant processes. Field and gate oxides are "hardened" to make the device less susceptible to radiation damage by modifying the process architecture to allow lower temperature processing. Device implants and Vts adjustments allow more Vt margin. In addition to process changes, IDT's radiation enhanced process utilizes epitaxial substrate material. The use of epi substrate material provides a lower substrate resistance environment to create latch-up free CMOS structures.

RADIATION HARDNESS CATEGORIES

Radiation Enhanced ('RE) or Radiation Tolerant ('RT) versions of IDT products follow IDT's military product data sheets whenever possible (consult factory). IDT's Total Dose Test plan exposes a sample of die on a wafer to a particular Total Dose level via ARACOR X-Ray radiation. This Total Dose Test plan qualifies each 'RE or 'RT wafer to a Total Dose level. Only wafers with sampled die that pass Total Dose level tests are assembled and used for orders (consult factory for more details on Total Dose sample testing). With regard to Total Dose testing, clarifications/exceptions to MIL-STD-883, Methods 5005 and 1019 are required. Consult factory for more details.

The 'RE and 'RT process enhancements enable IDT to offer integrated circuits with varying grades of radiation tolerance or radiation "hardness".

- Radiation Enhanced process uses Epi wafers and is able to provide devices that can be Total Dose qualified to 10K RADs (Si) or greater by IDT's ARACOR X-Ray Total Dose sample die test plan (Total Dose levels require negotiation, consult factory for more details).
- Radiation Tolerant product uses standard wafer/process material that is qualified to 10K RADs (Si) Total Dose by IDT's ARACOR X-Ray Total Dose sample die test plan.

Integrated Device Technology can provide Radiation Tolerant/Enhanced versions of all product types (some speed grades may not be available as 'RE).

Please contact your IDT sales representative or factory marketing to determine availability and price of any IDT product processed in accordance with one of these levels of radiation hardness.

CONCLUSION

There has been widespread interest within the military and space community in IDT's CMOS product line for its radiation hardness levels, as well as its high-performance and low power dissipation. To serve this growing need for CMOS circuits that must operate in a radiation environment, IDT has created a separate group within the company to concentrate on supplying products for these applications.Continuing research and development of process and products, including the use of in-house radiation testing capability, will allow Integrated Device Technology to offer continuously increasing levels of radiation-tolerant solutions.

THERMAL PERFORMANCE CALCULATIONS FOR IDT'S PACKAGES

Since most of the electrical energy consumed by microelectronic devices eventually appears as heat, poor thermal performance of the device or lack of management of this thermal energy can cause a variety of deleterious effects. This device temperature increase can exhibit itself as one of the key variables in establishing device performance and long term reliability; on the other hand, effective dissipation of internally generated thermal energy can, if properly managed, reduce the deleterious effects and improve component reliability.

A few key benefits of IDT's enhanced CEMOS™ process are: low power dissipation, high speed, increased levels of integration, wider operating temperature ranges and lower quiescent power dissipation. Because the reliability of an integrated circuit is largely dependent on the maximum temperature the device attains during operation, and as the junction stability declines with increases in junction temperature (T_J), it becomes increasingly important to maintain a low (T_J).

CMOS devices stabilize more quickly and at greatly lower temperature than bipolar devices under normal operation. The accelerated aging of an integrated circuit can be expressed as an exponential function of the junction temperature as:

$$t_A = t_o \exp\left[\frac{E_a}{k}\left(\frac{1}{T_O} - \frac{1}{T_J}\right)\right]$$

where

- t_A = lifetime at elevated junction (T_J) temperature
- t_O = normal lifetime at normal junction (T_O) temperature
- E_a = activation energy (ev)
- k = Boltzmann's constant (8.617×10^{-5} ev/k)

i.e. the lifetime of a device could be decreased by a factor of 2 for every 10°C increase temperature.

To minimize the deleterious effects associated with this potential increase, IDT has:

1. Optimized our proprietary low-power CEMOS fabrication process to ensure the active junction temperature rise is minimal.
2. Selected only packaging materials that optimize heat dissipation, which encourages a cooler running device.
3. Physically designed all package components to enhance the inherent material properties and to take full advantage of heat transfer and radiation due to case geometries.
4. Tightly controlled the assembly procedures to meet or exceed the stringent criteria of MIL-STD-883_to ensure maximum heat transfer between die and packaging materials.

The following figures graphically illustrate the thermal values of IDT's current package families. Each envelop (shaded area) depicts a typical spread of values due to the influence of a number of factors which include: circuit size, package cavity size and die attach integrity. The following range of values are to be used as a comprehensive characterization of the major variables rather than single point of reference.

When calculating junction temperature (T_J), it is necessary to know the thermal resistance of the package (θ_{JA}) as measured in "degree celsius per watt". With the accompanying data, the following equation can be used to establish thermal performance, enhance device reliability and ultimately provide you, the user, with a continuing series of high-speed, low-power CMOS solutions to your system design needs.

4

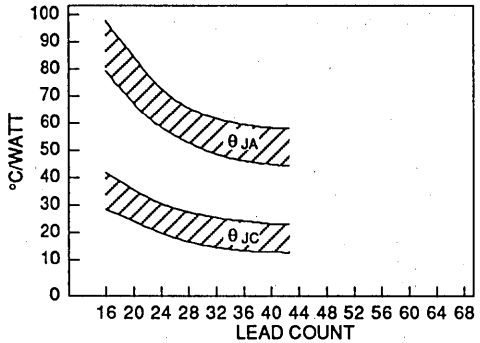
$$\theta_{JA} = [T_J - T_A]/P$$
$$T_J = T_A + P[\theta_{JA}] = T_A + P[\theta_{JC} + \theta_{CA}]$$

where

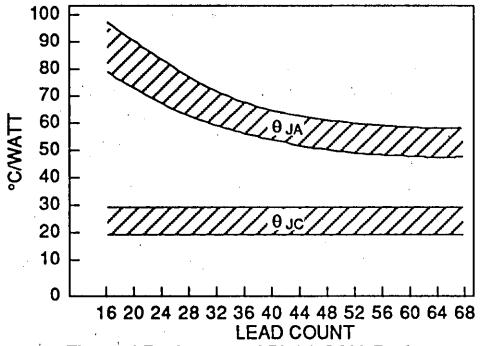
$$\theta_{JC} = \frac{T_J - T_C}{P} \qquad \theta_{CA} = \frac{T_C - T_A}{P}$$

- θ = Thermal resistance
- J = Junction
- P = Operational power of device (dissipated)
- T_A = Ambient temperature in degree celsius
- T_J = Temperature of the junction
- T_C = Temperature of case/package
- θ_{CA} = Case to Ambient, thermal resistance—usually a measure of the heat dissipation due to natural or forced convection, radiation and mounting techniques.
- θ_{JC} = Junction to Case, thermal resistance—usually measured with reference to the temperature at a specific point on the package (case) surface. (Dependent on the package material properties and package geometry.)
- θ_{JA} = Junction to Ambient, thermal resistance—usually measured with respect to the temperature of a specified volume of still air. (Dependent on θ_{JC} + θ_{JA} which includes the influence of area and environmental condition.)

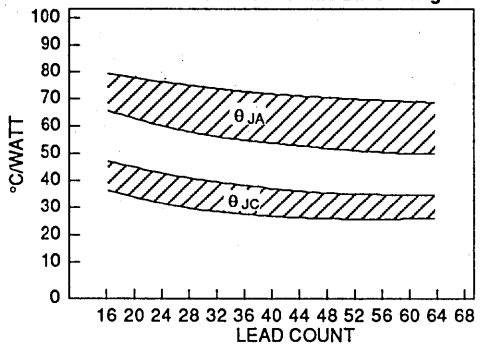
Ref. MIL-STD-883C, Method 1012.1
JEDEC ENG. Bulletin No. 20, January 1975
1986 Semi. Std., Vol. 4, Test Methods G30–86, G32–86.



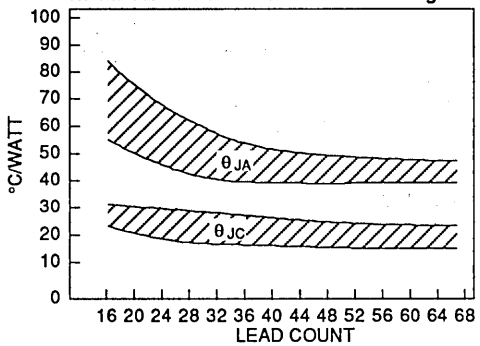
Thermal Resistance of Ceramic DIP Packages



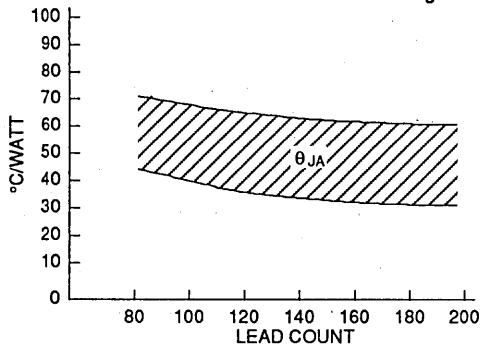
Thermal Resistance of PLCC/SOIC Packages



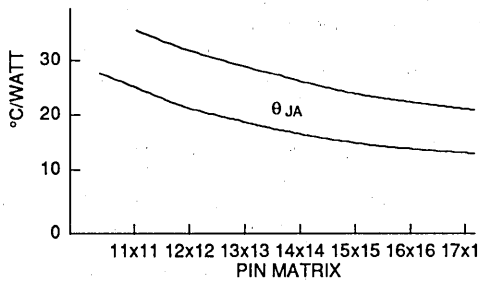
Thermal Resistance of Plastic DIP Packages



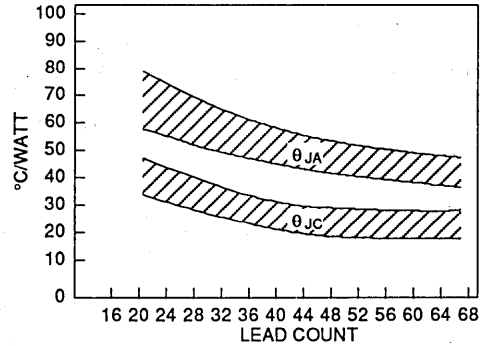
Thermal Resistance of Ceramic Sidebraze Packages



Thermal Resistance of PPGA Packages



PGA Thermal Resistance



Thermal Resistance of Ceramic Leadless Chip Carrier (LCC) Packages

2512 drw 01

PACKAGE DIAGRAM OUTLINE INDEX

MONOLITHIC PACKAGE DIAGRAM OUTLINES (Continued)

PAGE

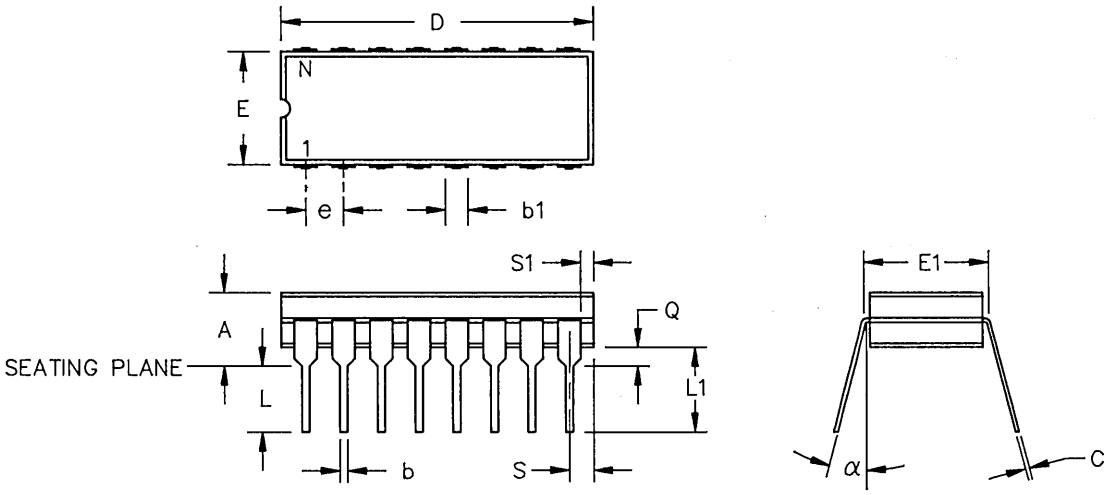
MONOLITHIC PACKAGE DIAGRAM OUTLINES (Continued)

4

Integrated Device Technology, Inc.

PACKAGE DIAGRAM OUTLINES

DUAL IN–LINE PACKAGES



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. THE MINIMUM LIMIT FOR DIMENSION b1 MAY BE .023 FOR CORNER LEADS.

16–28 LEAD CERDIP (300 MIL)

DWG #	D16–1		D18–1		D20–1		D22–1		D24–1		D28–3	
# OF LDS (N)	16		18		20		22		24		28	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.140	.200	.140	.200	.140	.200	.140	.200	.140	.200	.140	.200
b	.015	.021	.015	.021	.015	.021	.015	.021	.015	.021	.015	.021
b1	.038	.060	.038	.060	.038	.060	.045	.060	.045	.065	.045	.065
C	.009	.012	.009	.012	.009	.012	.009	.012	.009	.014	.009	.014
D	.750	.830	.880	.930	.935	1.060	1.050	1.080	1.240	1.280	1.440	1.490
E	.285	.310	.285	.310	.285	.310	.285	.310	.285	.310	.285	.310
E1	.290	.320	.290	.320	.290	.320	.300	.320	.300	.320	.300	.320
e	.100 BSC		.100 BSC		.100 BSC		.100 BSC		.100 BSC		.100 BSC	
L	.125	.175	.125	.175	.125	.175	.125	.175	.125	.175	.125	.175
L1	.150	–	.150	–	.150	–	.150	–	.150	–	.150	–
Q	.015	.055	.015	.055	.015	.060	.015	.060	.015	.060	.015	.060
S	.020	.080	.020	.080	.020	.080	.020	.080	.030	.080	.030	.080
S1	.005	–	.005	–	.005	–	.005	–	.005	–	.005	–
α	0°	15°	0°	15°	0°	15°	0°	15°	0°	15°	0°	15°

DUAL IN-LINE PACKAGES (Continued)

24-40 LEAD CERDIP (600 MIL)

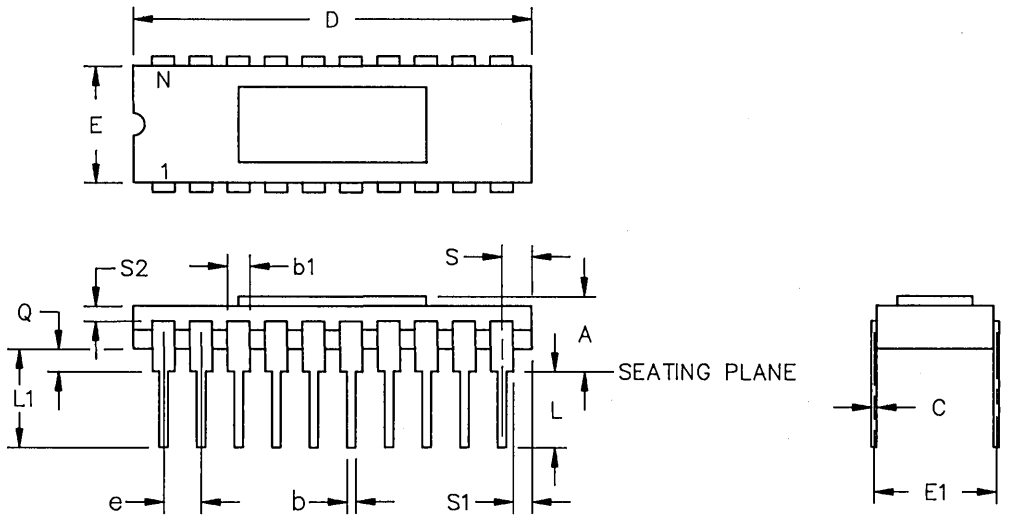
DWG #	D24-2		D28-1		D40-1	
# OF LDS (N)	24		28		40	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX
A	.090	.190	.090	.200	.160	.220
b	.014	.023	.014	.023	.014	.023
b1	.038	.060	.038	.065	.038	.065
C	.008	.012	.008	.014	.008	.014
D	1.230	1.290	1.440	1.490	2.020	2.070
E	.500	.610	.510	.545	.510	.545
E1	.590	.620	.590	.620	.590	.620
e	.100 BSC		.100 BSC		.100 BSC	
L	.125	.200	.125	.200	.125	.200
L1	.150	–	.150	–	.150	–
Q	.015	.060	.020	.060	.020	.060
S	.030	.080	.030	.080	.030	.080
S1	.005	–	.005	–	.005	–
α	0°	15°	0°	15°	0°	15°

28-40 LEAD CERDIP (WIDE BODY)

DWG #	D28-2		D32-1		D40-2	
# OF LDS (N)	28		32		40	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX
A	.090	.200	.120	.210	.160	.220
b	.014	.023	.014	.023	.014	.023
b1	.038	.065	.038	.065	.038	.065
C	.008	.014	.008	.014	.008	.014
D	1.440	1.490	1.625	1.675	2.020	2.070
E	.570	.600	.570	.600	.570	.600
E1	.590	.620	.590	.620	.590	.620
e	.100 BSC		.100 BSC		.100 BSC	
L	.125	.200	.125	.200	.125	.200
L1	.150	–	.150	–	.150	–
Q	.020	.060	.020	.060	.020	.060
S	.030	.080	.030	.080	.030	.080
S1	.005	–	.005	–	.005	–
α	0°	15°	0°	15°	0°	15°

DUAL IN-LINE PACKAGES (Continued)

20-32 LEAD SIDE BRAZE (300 MIL)

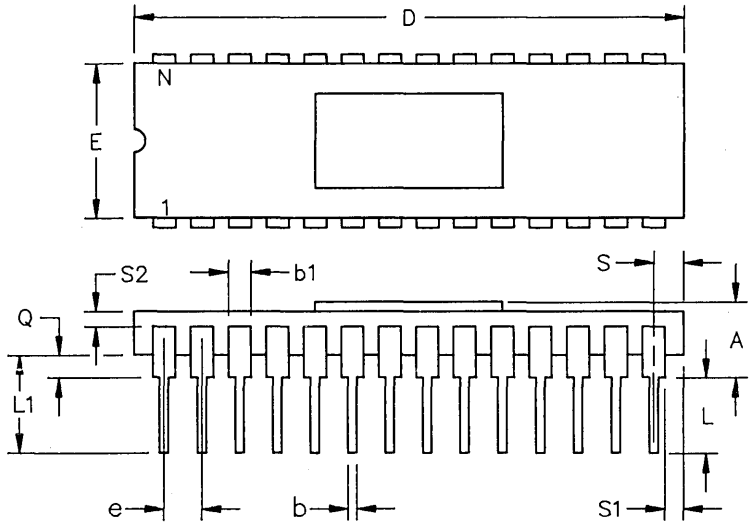


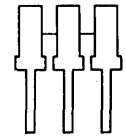
NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	C20-1		C22-1		C24-1		C28-1		C32-3	
# OF LDS (N)	20		22		24		28		32	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.090	.200	.100	.200	.090	.200	.090	.200	.090	.200
b	.014	.023	.014	.023	.015	.023	.014	.023	.014	.023
b1	.040	.060	.040	.060	.040	.060	.040	.060	.040	.060
C	.008	.015	.008	.015	.008	.015	.008	.015	.008	.014
D	.970	1.060	1.040	1.120	1.180	1.230	1.380	1.420	1.580	1.640
E	.260	.310	.260	.310	.220	.310	.220	.310	.280	.310
E1	.290	.320	.290	.320	.290	.320	.290	.320	.290	.320
e	.100 BSC		.100 BSC		.100 BSC		.100 BSC		.100 BSC	
L	.125	.200	.125	.200	.125	.200	.125	.200	.100	.175
L1	.150	–	.150	–	.150	–	.150	–	.150	–
Q	.015	.060	.015	.060	.015	.060	.015	.060	.030	.060
S	.030	.065	.030	.065	.030	.065	.030	.065	.030	.065
S1	.005	–	.005	–	.005	–	.005	–	.005	–
S2	.005	–	.005	–	.005	–	.005	–	.005	–

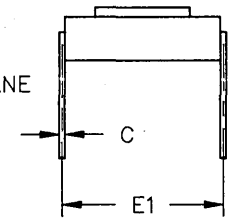
DUAL IN-LINE PACKAGES (Continued)

28-48 LEAD SIDE BRAZE (400 MIL)





48 LEAD OPTION

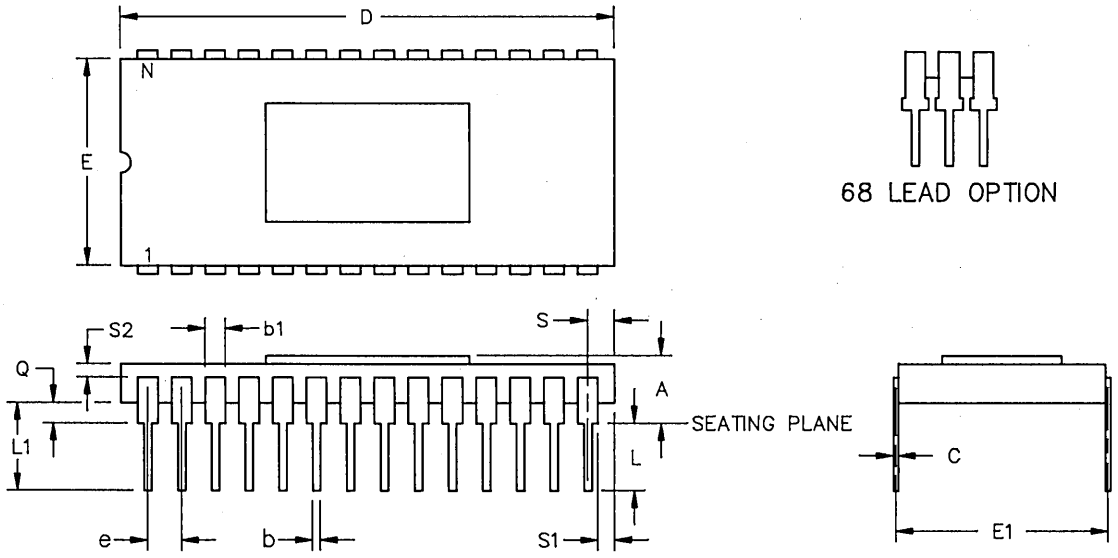


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	C28-2		C32-2		C48-1	
# OF LDS (N)	28		32		48	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX
A	.090	.200	.090	.200	.085	.190
b	.014	.023	.014	.023	.014	.023
b1	.040	.060	.040	.060	.040	.060
C	.008	.014	.008	.014	.008	.014
D	1.380	1.420	1.580	1.640	1.690	1.730
E	.380	.420	.380	.410	.380	.410
E1	.390	.420	.390	.420	.390	.420
e	.100 BSC		.100 BSC		.070 BSC	
L	.100	.175	.100	.175	.125	.175
L1	.150	–	.150	–	.150	–
Q	.030	.060	.030	.060	.020	.070
S	.030	.065	.030	.065	.030	.065
S1	.005	–	.005	–	.005	–
S2	.005	–	.005	–	.005	–

DUAL IN-LINE PACKAGES (Continued)

24-68 LEAD SIDE BRAZE (600 MIL)

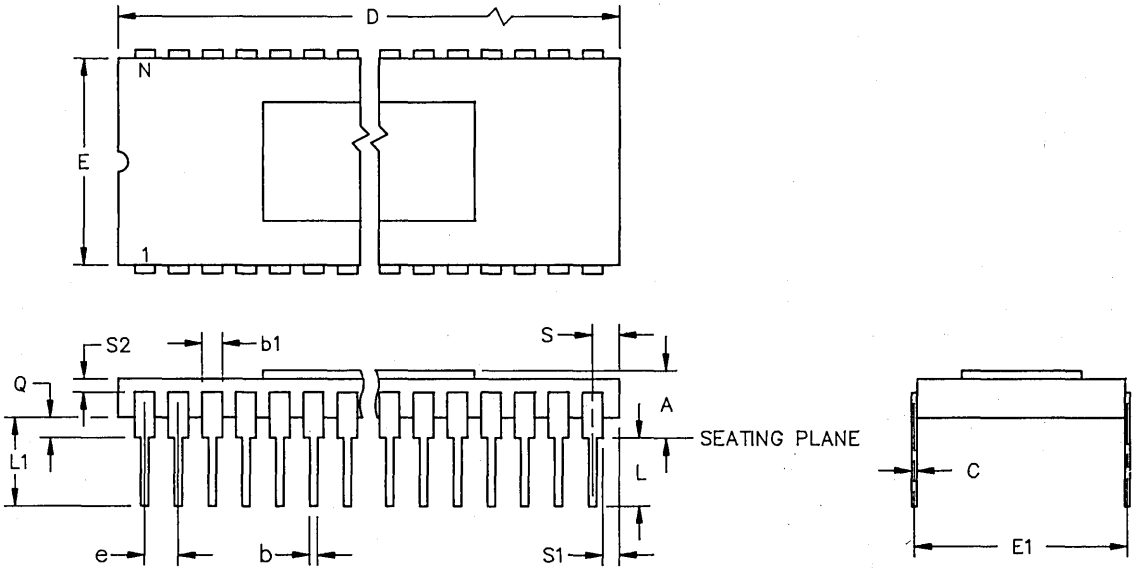


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	C24-2		C28-3		C32-1		C40-1		C48-2		C68-1	
# OF LDS (N)	24		28		32		40		48		68	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.090	.190	.085	.190	.100	.190	.085	.190	.100	.190	.085	.190
b	.015	.023	.015	.022	.015	.023	.015	.023	.015	.023	.015	.023
b1	.040	.060	.038	.060	.040	.060	.038	.060	.040	.060	.040	.060
C	.008	.012	.008	.012	.008	.014	.008	.012	.008	.012	.008	.012
D	1.180	1.220	1.380	1.430	1.580	1.640	1.980	2.030	2.370	2.430	2.380	2.440
E	.575	.610	.580	.610	.580	.610	.580	.610	.550	.610	.580	.610
E1	.595	.620	.595	.620	.590	.620	.595	.620	.595	.620	.590	.620
e	.100 BSC		.100 BSC		.100 BSC		.100 BSC		.100 BSC		.070 BSC	
L	.125	.175	.125	.175	.100	.175	.125	.175	.125	.175	.125	.175
L1	.150	–	.150	–	.150	–	.150	–	.150	–	.150	–
Q	.020	.060	.020	.065	.020	.060	.020	.060	.020	.060	.020	.070
S	.030	.065	.030	.065	.030	.065	.030	.065	.030	.065	.030	.065
S1	.005	–	.005	–	.005	–	.005	–	.005	–	.005	–
S2	.005	–	.005	–	.005	–	.005	–	.005	–	.005	–

DUAL IN-LINE PACKAGES (Continued)

64 LEAD SIDE BRAZE (900 MIL)

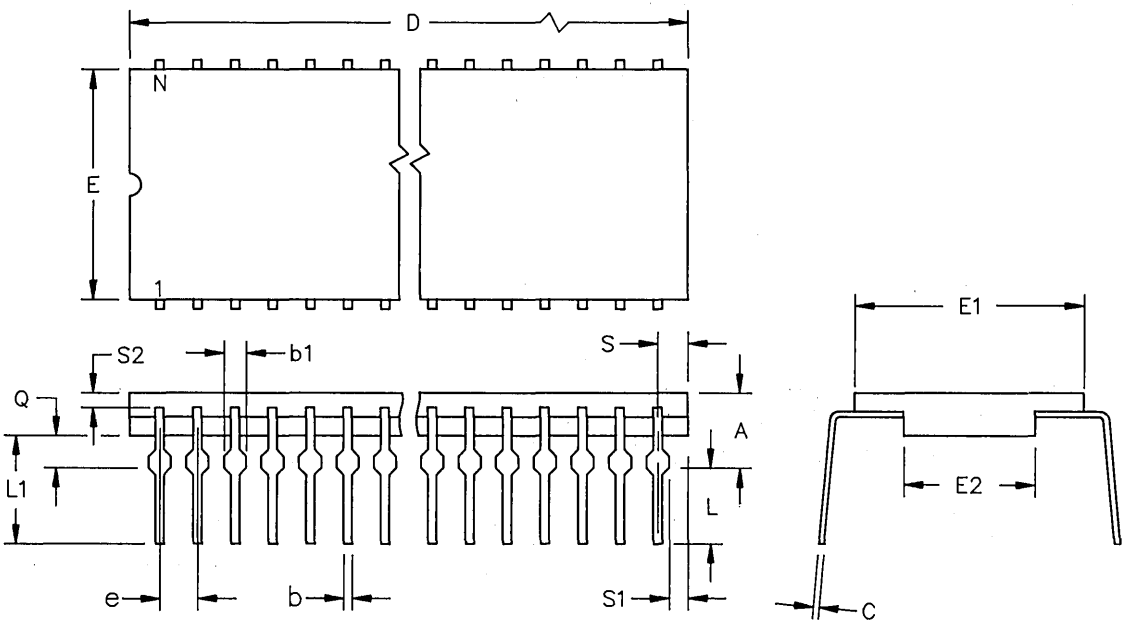


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	C64-1	
# OF LDS (N)	64	
SYMBOL	MIN	MAX
A	.110	.190
b	.014	.023
b1	.040	.060
C	.008	.015
D	3.160	3.240
E	.884	.915
E1	.890	.920
e	.100 BSC	
L	.125	.200
L1	.150	–
Q	.015	.070
S	.030	.065
S1	.005	–
S2	.005	–

DUAL IN-LINE PACKAGES (Continued)

64 LEAD TOP BRAZE (900 MIL)

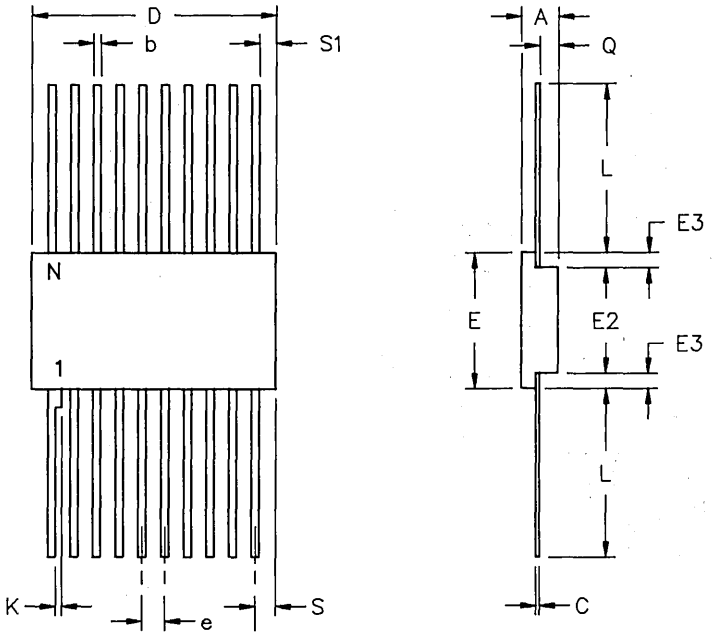


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	C64-2	
# OF LDS (N)	64	
SYMBOL	MIN	MAX
A	.120	.180
b	.015	.021
b1	.040	.060
C	.009	.012
D	3.170	3.240
E	.790	.810
E1	.880	.815
E2	.640	.660
e	.100 BSC	
L	.125	.160
L1	.150	–
Q	.020	.100
S	.030	.065
S1	.005	–
S2	.005	–

FLATPACKS

20–28 LEAD FLATPACK

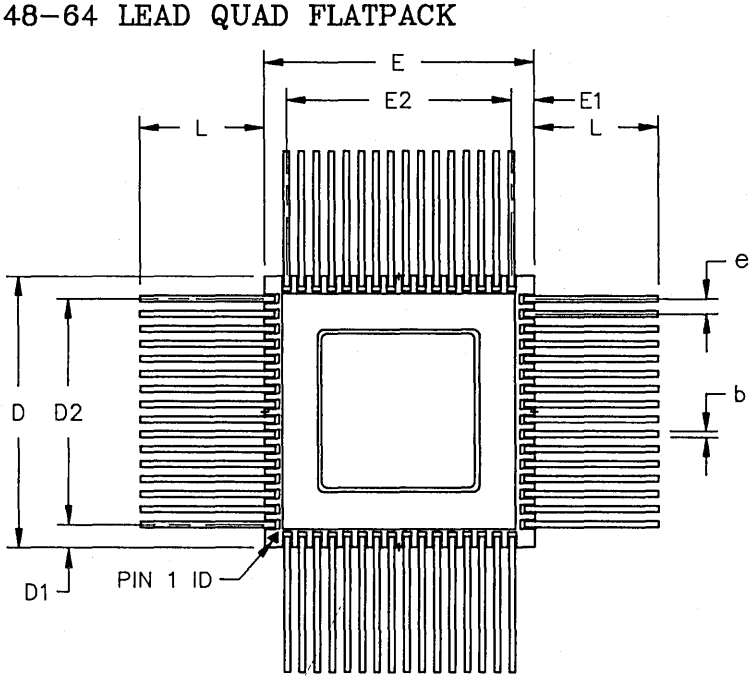


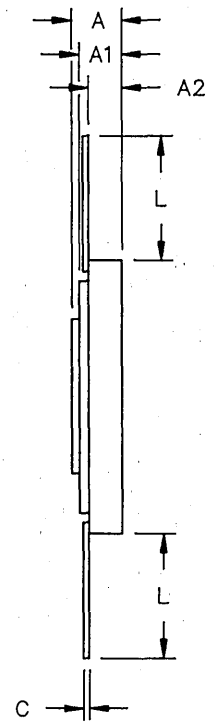
NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	F20–1		F20–2		F24–1		F28–1		F28–2	
# OF LDS (N)	20		20 (.295 BODY)		24		28		28	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.045	.092	.045	.092	.045	.090	.045	.090	.045	.115
b	.015	.019	.015	.019	.015	.019	.015	.019	.015	.019
C	.003	.006	.003	.006	.003	.006	.004	.007	.003	.007
D	–	.540	–	.540	–	.640	.710	.740	.710	.740
E	.340	.360	.245	.303	.360	.420	.480	.520	.480	.520
E2	.130	–	.130	–	.180	–	.180	–	.180	–
E3	.030	–	.030	–	.030	–	.040	–	.040	–
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
K	.006	.015	.008	.015	–	–	–	–	–	–
L	.250	.370	.250	.370	.250	.370	.250	.370	.250	.370
Q	.010	.040	.010	.040	.010	.040	.010	.045	.026	.045
S	–	.045	–	.045	–	.045	–	.045	–	.045
S1	.000	–	.005	–	.005	–	.005	–	.005	–

FLATPACKS (Continued)

48–64 LEAD QUAD FLATPACK



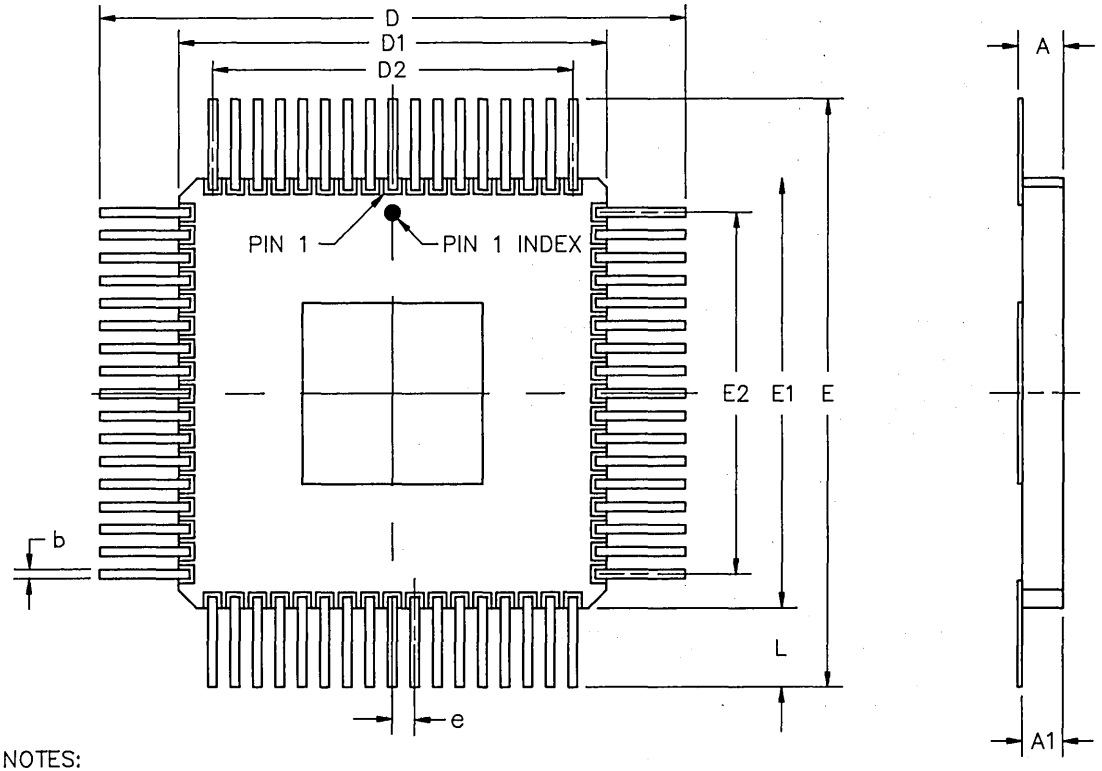


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	F48–1		F64–1	
# OF LDS (N)	48		64	
SYMBOL	MIN	MAX	MIN	MAX
A	.089	.108	.070	.090
A1	.079	.096	.060	.078
A2	.058	.073	.030	.045
b	.018	.022	.016	.020
C	.008	.010	.009	.012
D/E	–	.750	.885	.915
D1/E1	.100 REF		.075 REF	
D2/E2	.550 BSC		.750 BSC	
e	.050 BSC		.050 BSC	
L	.350	.450	.350	.450
ND/NE	12		16	

FLATPACKS (Continued)

68 LEAD QUAD FLATPACK

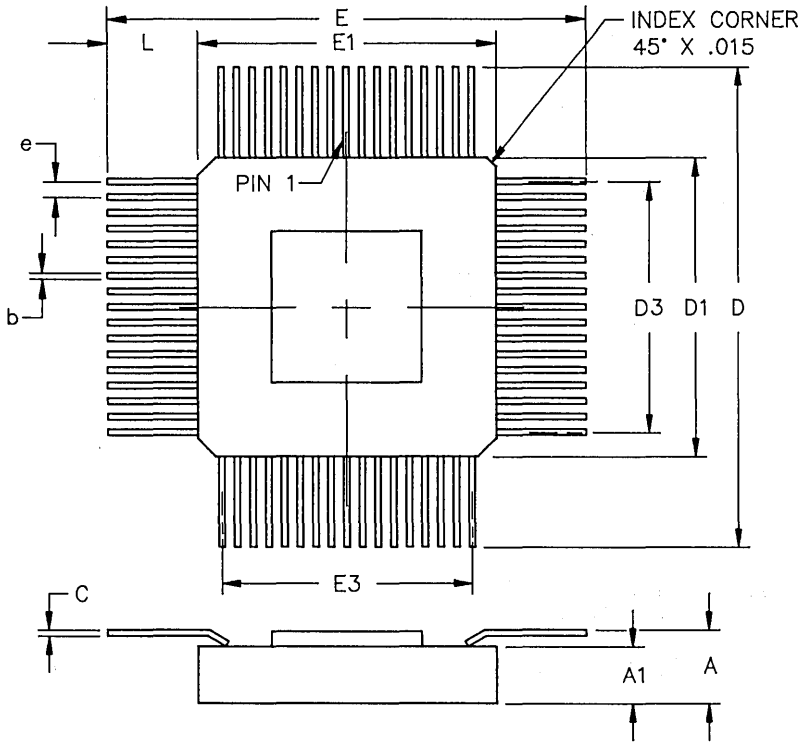


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	F68–1	
# OF LDS (N)	68	
SYMBOL	MIN	MAX
A	.080	.145
A1	.070	.090
b	.014	.021
C	.008	.012
D/E	1.640	1.870
D1/E1	.926	.970
D2/E2	.800 BSC	
e	.050 BSC	
L	.350	.450
ND/NE	17	

FLATPACKS (Continued)

68 LEAD QUAD FLATPACK (STRAIGHT LEADS)

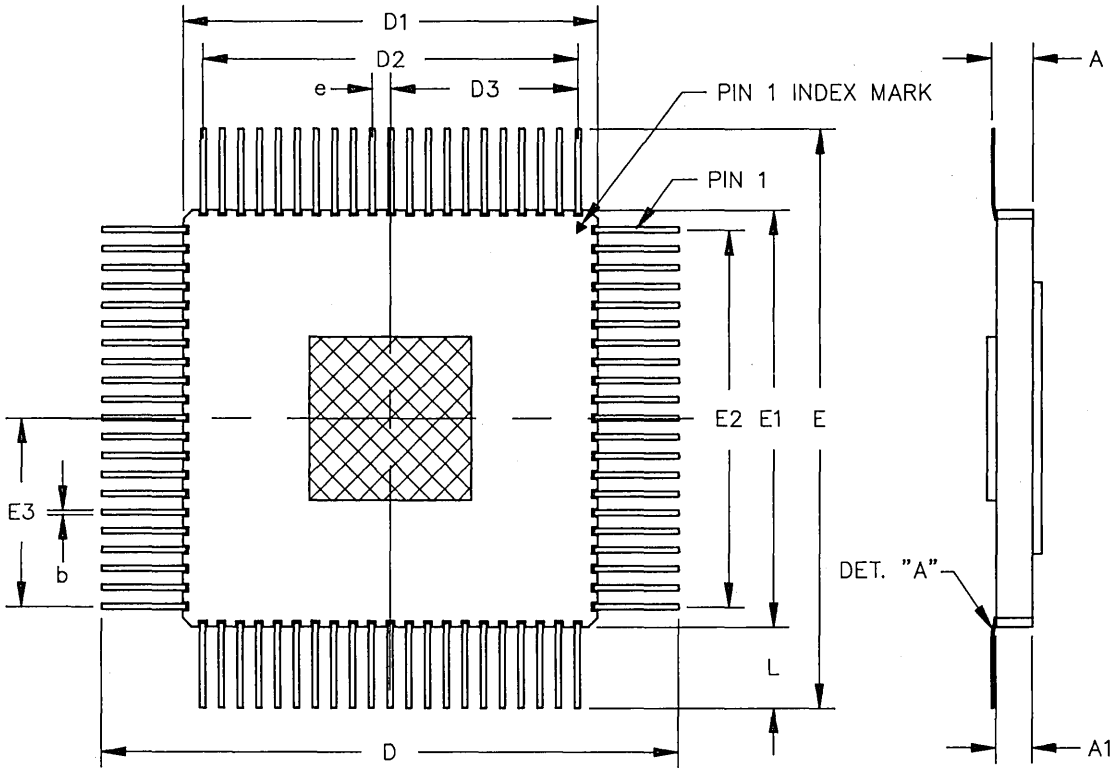


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	F68–2	
# OF LDS (N)	68	
SYMBOL	MIN	MAX
A	.064	.084
A1	.054	.070
b	.008	.013
C	.0045	.008
D/E	.860	1.100
D1/E1	.460	.500
D2/E2	.400 REF	
e	.025 BSC	
L	.200	.300
ND/NE	17	

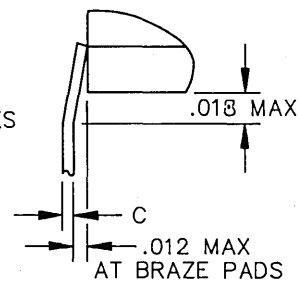
FLATPACKS (Continued)

84 LEAD QUAD FLATPACK (CAVITY DOWN)



DWG #	F84–1	
# OF LDS (N)	84	
SYMBOL	MIN	MAX
A	–	.140
A1	–	.105
b	.014	.020
C	.007	.013
D/E	1.485	1.615
D1/E1	1.130	1.170
D2/E2	1.000 BSC	
D3/E3	.500 BSC	
e	.050 BSC	
L	.350	.450
ND/NE	21	

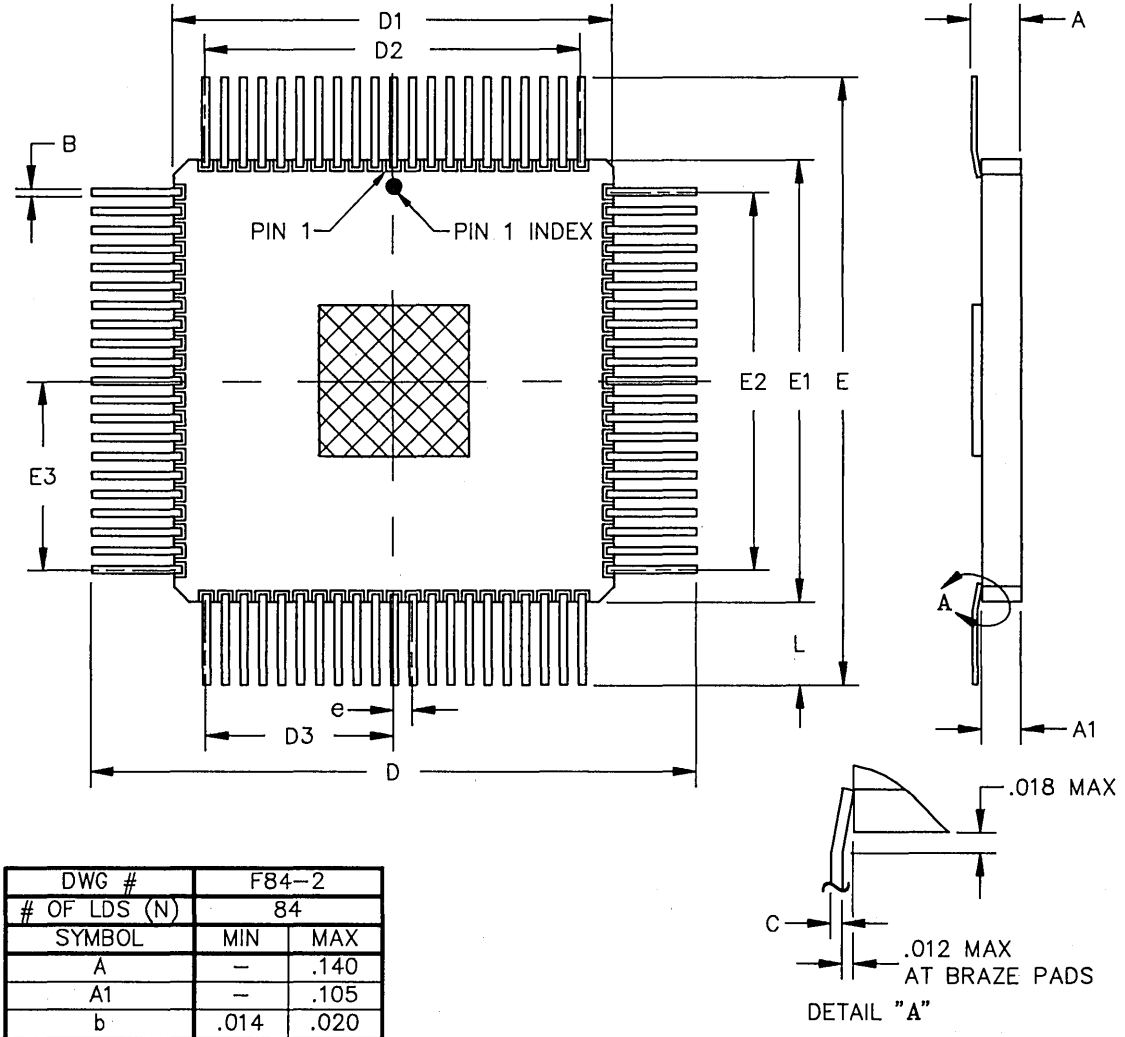
NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. CROSS HATCHED AREA INDICATES INTEGRAL METALLIC HEAT SINK.



DETAIL A

FLATPACKS (Continued)

84 LEAD QUAD FLATPACK (CAVITY UP)

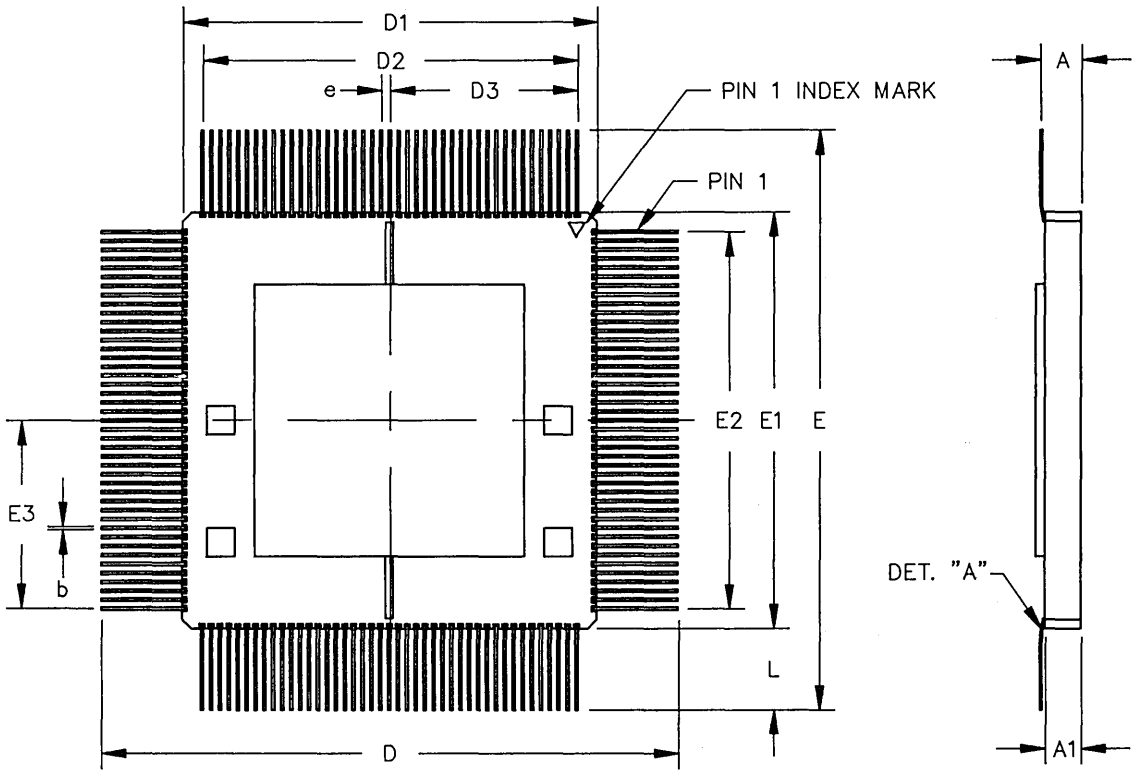


DWG #	F84-2	
# OF LDS (N)	84	
SYMBOL	MIN	MAX
A	–	.140
A1	–	.105
b	.014	.020
C	.007	.013
D/E	1.485	1.615
D1/E1	1.130	1.170
D2/E2	1.000 BSC	
D3/E3	.500 BSC	
e	.050 BSC	
L	.350	.450
ND/NE	21	

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. CROSS HATCHED AREA INDICATES INTEGRAL METALLIC HEAT SINK.

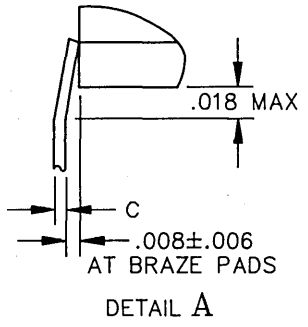
FLATPACKS (Continued)

172 LEAD QUAD FLATPACK (CAVITY UP – R3001)



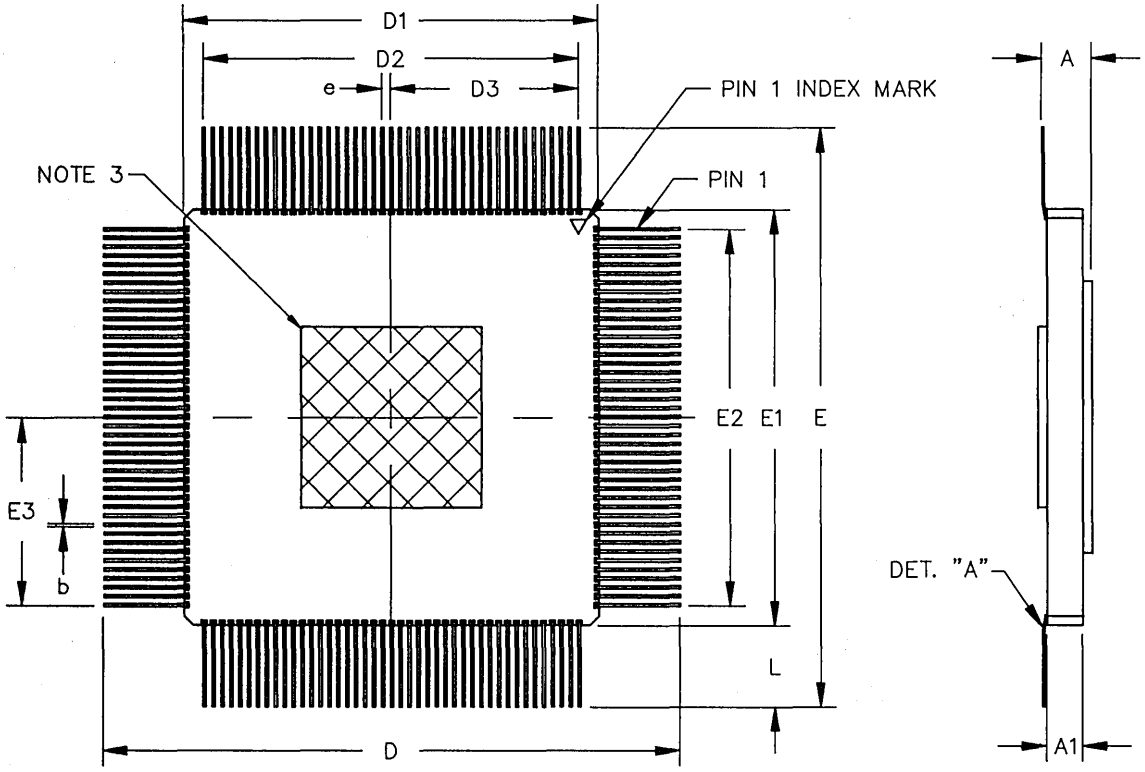
DWG #	F172–1	
# OF LDS (N)	172	
SYMBOL	MIN	MAX
A	–	.130
A1	–	.105
b	.006	.010
C	.004	.008
D/E	1.580	1.620
D1/E1	1.135	1.165
D2/E2	1.050 BSC	
D3/E3	.525 BSC	
e	.025 BSC	
L	.220	.230
ND/NE	43	

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.



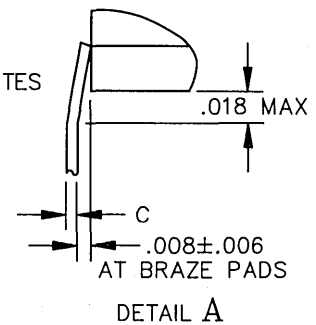
FLATPACKS (Continued)

172 LEAD QUAD FLATPACK (CAVITY DOWN – R3000A)



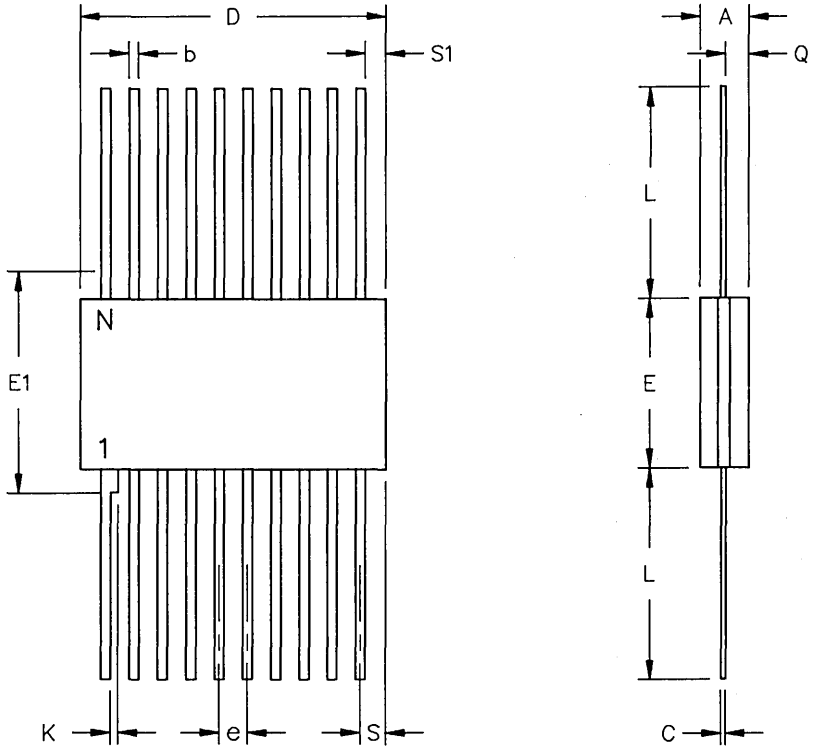
DWG #	F172–2	
# OF LDS (N)	172	
SYMBOL	MIN	MAX
A	–	.130
A1	–	.105
b	.006	.010
C	.004	.008
D/E	1.580	1.620
D1/E1	1.135	1.165
D2/E2	1.050 BSC	
D3/E3	.525 BSC	
e	.025 BSC	
L	.220	.230
ND/NE	43	

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. CROSS HATCHED AREA INDICATES METALLIC HEAT SINK.



CERPACKS

16–28 LEAD CERPACK

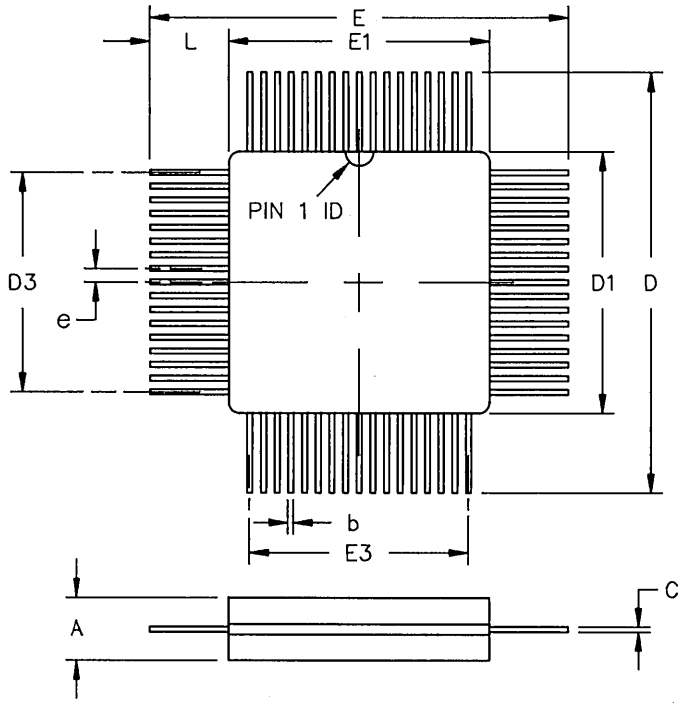


NOTES:
1. ALL DIMENSION ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	E16–1		E20–1		E24–1		E28–1		E28–2	
# OF LDS (N)	16		20		24		28		28	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.055	.085	.045	.092	.045	.090	.045	.115	.045	.090
b	.015	.019	.015	.019	.015	.019	.015	.019	.015	.019
C	.0045	.006	.0045	.006	.0045	.006	.0045	.009	.0045	.006
D	.370	.430	–	.540	–	.640	–	.740	–	.740
E	.245	.285	.245	.300	.300	.420	.460	.520	.340	.380
E1	–	.305	–	.305	–	.440	–	.550	–	.400
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
K	.008	.015	.008	.015	.008	.015	.008	.015	.008	.015
L	.250	.370	.250	.370	.250	.370	.250	.370	.250	.370
Q	.026	.040	.026	.040	.026	.040	.026	.045	.026	.045
S	–	.045	–	.045	–	.045	–	.045	–	.045
S1	.005	–	.005	–	.005	–	.000	–	.005	–

CERQUADS

68 LEAD CERQUAD (STRAIGHT LEADS)

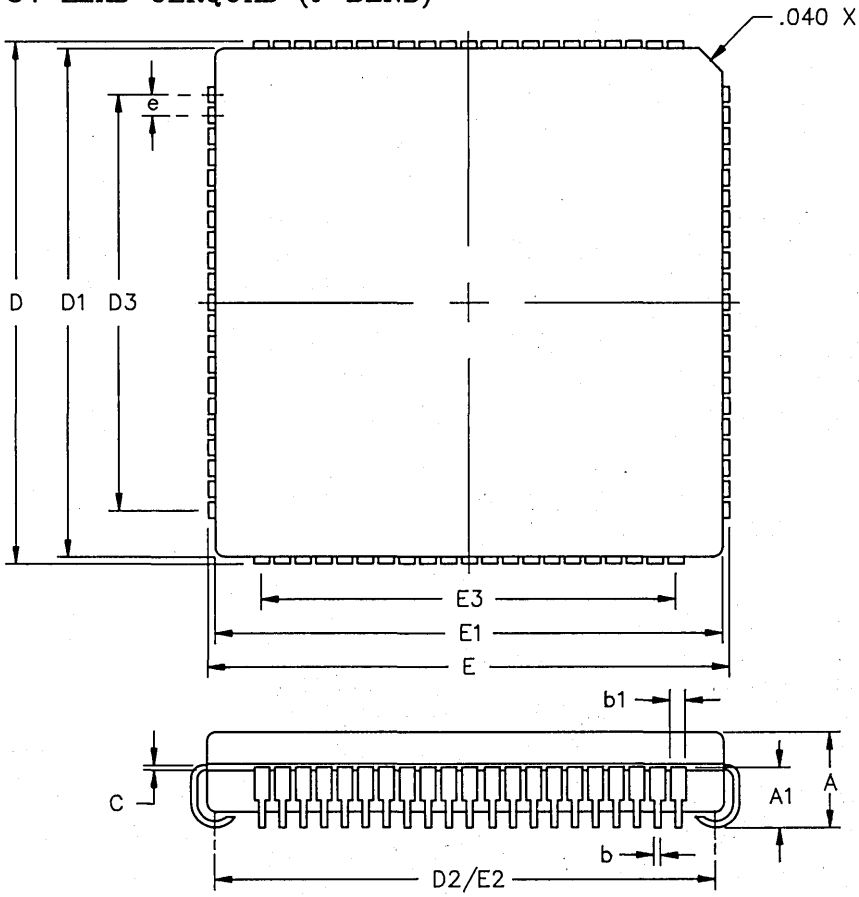


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

DWG #	CQ68–1	
# OF LDS (N)	68	
SYMBOL	MIN	MAX
A	.115	.165
b	.008	.013
C	.0045	.008
D/E	.860	1.100
D1/E1	.460	.500
D3/E3	.400 REF	
e	.025 BSC	
L	.200	.300
ND/NE	17	

CERQUADS (Continued)

84 LEAD CERQUAD (J–BEND)

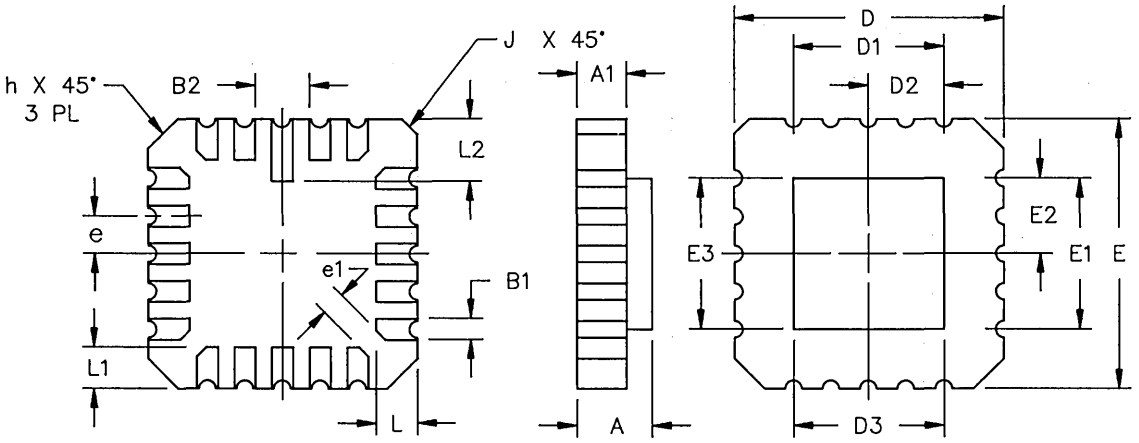


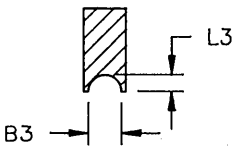
DWG #	CQ84–1	
# OF LDS (N)	84	
SYMBOL	MIN	MAX
A	.155	.200
A1	.090	.120
b1	.022	.032
b	.013	.023
C	.006	.013
D/E	1.170	1.190
D1/E1	1.138	1.162
D2/E2	1.100	.1.150
D3/.E3	1.000 BSC	
e	.050 BSC	
ND/NE	21	

NOTES:

1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

LEADLESS CHIP CARRIERS





NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

20–48 LEAD LCC (SQUARE)

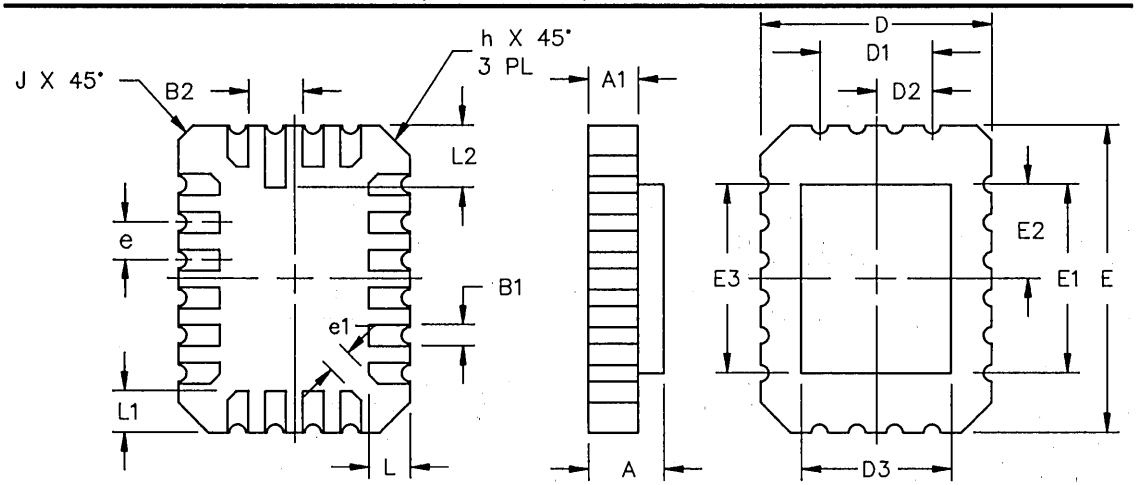
DWG #	L20–2		L28–1		L44–1		L48–1	
# OF LDS (N)	20		28		44		48	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.064	.100	.064	.100	.064	.120	.055	.120
A1	.054	.066	.050	.088	.054	.088	.045	.090
B1	.022	.028	.022	.028	.022	.028	.017	.023
B2	.072 REF		.072 REF		.072 REF		.072 REF	
B3	.006	.022	.006	.022	.006	.022	.006	.022
D/E	.342	.358	.442	.460	.640	.660	.554	.572
D1/E1	.200 BSC		.300 BSC		.500 BSC		.440 BSC	
D2/E2	.100 BSC		.150 BSC		.250 BSC		.220 BSC	
D3/E3	–	.358	–	.460	–	.560	.500	.535
e	.050 BSC		.050 BSC		.050 BSC		.040 BSC	
e1	.015	–	.015	–	.015	–	.015	–
h	.040 REF		.040 REF		.040 REF		.012 RADIUS	
J	.020 REF		.020 REF		.020 REF		.020 REF	
L	.045	.055	.045	.055	.045	.055	.033	.047
L1	.045	.055	.045	.055	.045	.055	.033	.047
L2	.077	.093	.077	.093	.077	.093	.077	.093
L3	.003	.015	.003	.015	.003	.015	.003	.015
ND/NE	5		7		11		12	

LEADLESS CHIP CARRIERS (Continued)

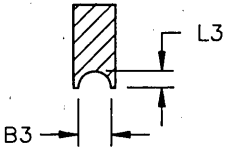
52–68 LEAD LCC (SQUARE)

DWG #	L52–1		L52–2		L68–2		L68–1	
# OF LDS (N)	52		52		68		68	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.061	.087	.082	.120	.082	.120	.065	.120
A1	.051	.077	.072	.088	.072	.088	.055	.075
B1	.022	.028	.022	.028	.022	.028	.008	.014
B2	.072 REF		.072 REF		.072 REF		.072 REF	
B3	.006	.022	.006	.022	.006	.022	.006	.022
D/E	.739	.761	.739	.761	.938	.962	.554	.566
D1/E1	.600 BSC		.600 BSC		.800 BSC		.400 BSC	
D2/E2	.300 BSC		.300 BSC		.400 BSC		.200 BSC	
D3/E3	–	.661	–	.661	–	.862	–	.535
e	.050 BSC		.050 BSC		.050 BSC		.025 BSC	
e1	.015	–	.015	–	.015	–	.015	–
h	.040 REF		.040 REF		.040 REF		.040 REF	
J	.020 REF		.020 REF		.020 REF		.020 REF	
L	.045	.055	.045	.055	.045	.055	.045	.055
L1	.045	.055	.045	.055	.045	.055	.045	.055
L2	.077	.093	.075	.093	.077	.093	.077	.093
L3	.003	.015	.003	.015	.003	.015	.003	.015
ND/NE	13		13		17		17	

LEADLESS CHIP CARRIERS (Continued)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.

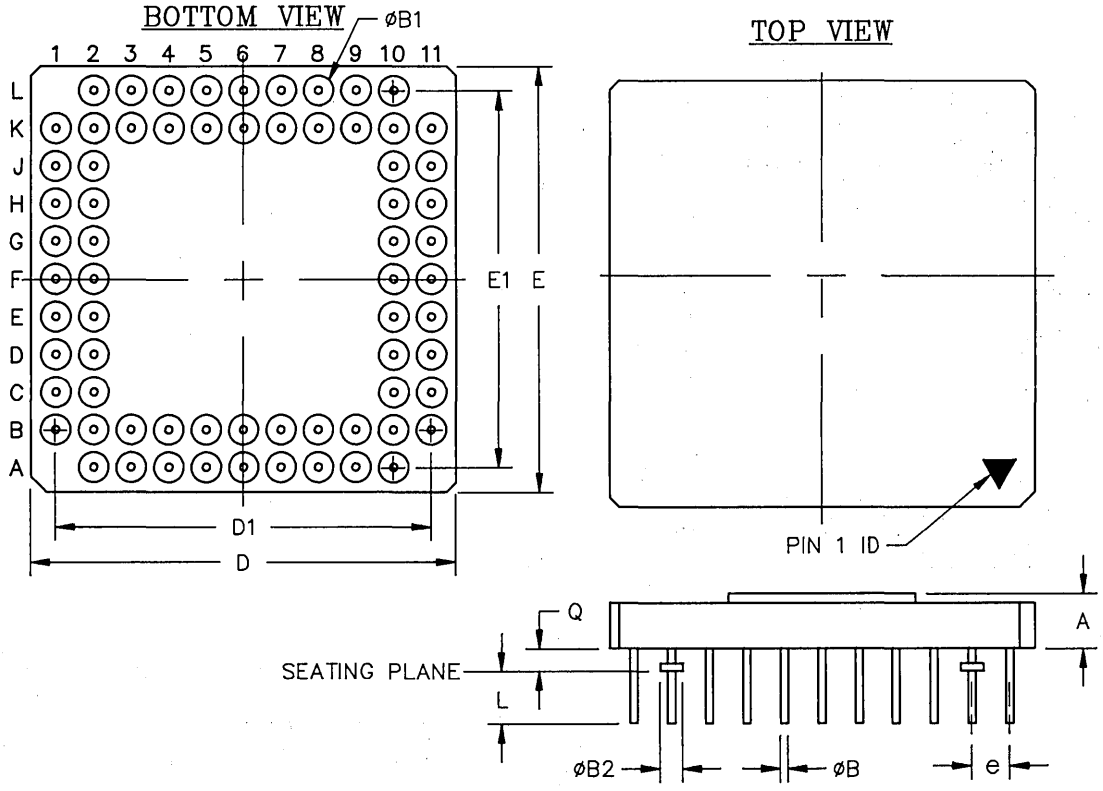


20–32 LEAD LCC (RECTANGULAR)

DWG #	L20–1		L22–1		L24–1		L28–2		L32–1	
# OF LDS (N)	20		22		24		28		32	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.060	.075	.064	.100	.064	.120	.060	.120	.060	.120
A1	.050	.065	.054	.063	.054	.066	.050	.088	.050	.088
B1	.022	.028	.022	.028	.022	.028	.022	.028	.022	.028
B2	.072 REF		.072 REF		.072 REF		.072 REF		.072 REF	
B3	.006	.022	.006	.022	.006	.022	.006	.022	.006	.022
D	.284	.296	.284	.296	.292	.308	.342	.358	.442	.458
D1	.150 BSC		.150 BSC		.200 BSC		.200 BSC		.300 BSC	
D2	.075 BSC		.075 BSC		.100 BSC		.100 BSC		.150 BSC	
D3	–	.280	–	.280	–	.308	–	.358	–	.458
E	.420	.435	.480	.496	.392	.408	.540	.560	.540	.560
E1	.250 BSC		.300 BSC		.300 BSC		.400 BSC		.400 BSC	
E2	.125 BSC		.150 BSC		.150 BSC		.200 BSC		.200 BSC	
E3	–	.410	–	.480	–	.408	–	.558	–	.558
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
e1	.015	–	.015	–	.015	–	.015	–	.015	–
h	.040 REF		.012 RADIUS		.025 REF		.040 REF		.040 REF	
J	.020 REF		.012 RADIUS		.015 REF		.020 REF		.020 REF	
L	.045	.055	.039	.051	.040	.050	.045	.055	.045	.055
L1	.045	.055	.039	.051	.040	.050	.045	.055	.045	.055
L2	.080	.095	.083	.097	.077	.093	.077	.093	.077	.093
L3	.003	.015	.003	.015	.003	.015	.003	.015	.003	.015
ND	4		4		5		5		7	
NE	6		7		7		9		9	

PIN GRID ARRAYS

68 PIN PGA (CAVITY UP)

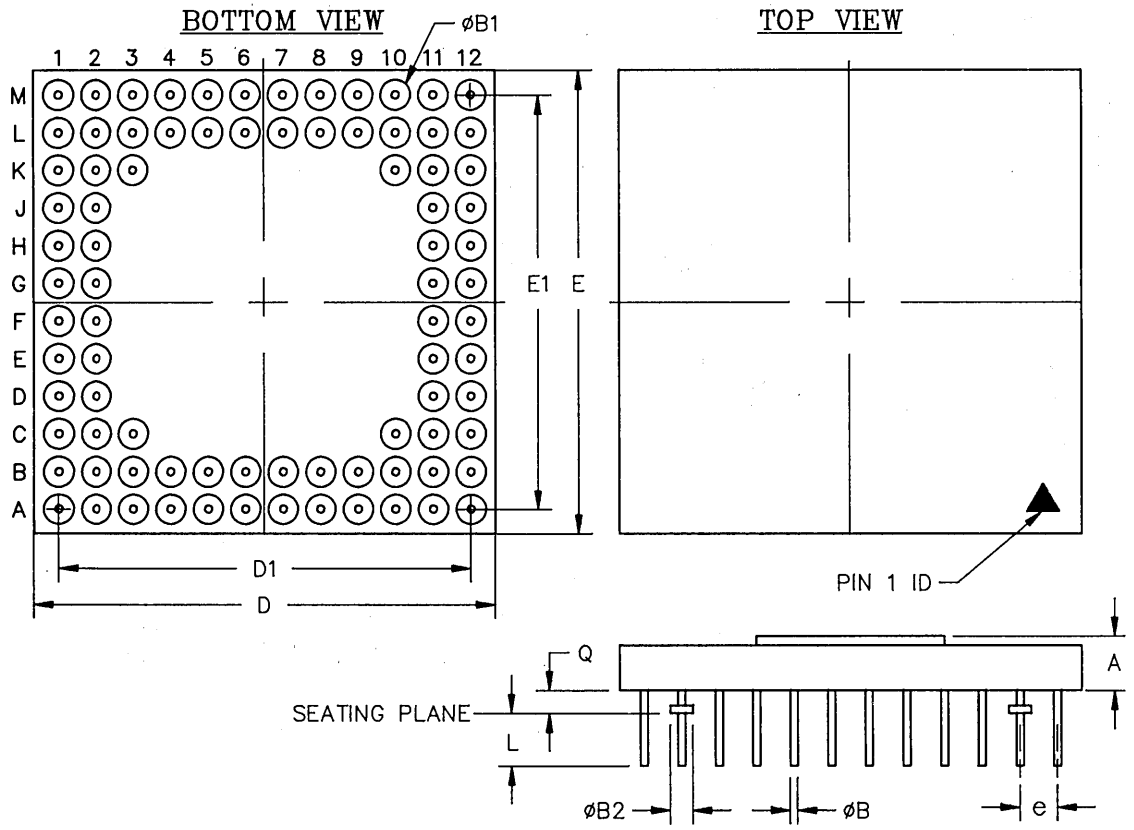


DWG #	G68-1	
# OF PINS (N)	68	
SYMBOL	MIN	MAX
A	.070	.145
øB	.016	.020
øB1	–	.080
øB2	.040	.060
D/E	1.140	1.180
D1/E1	1.000 BSC	
e	.100 BSC	
L	.120	.140
M	11	
Q	.040	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

84 PIN PGA (CAVITY UP – 12 X 12 GRID)

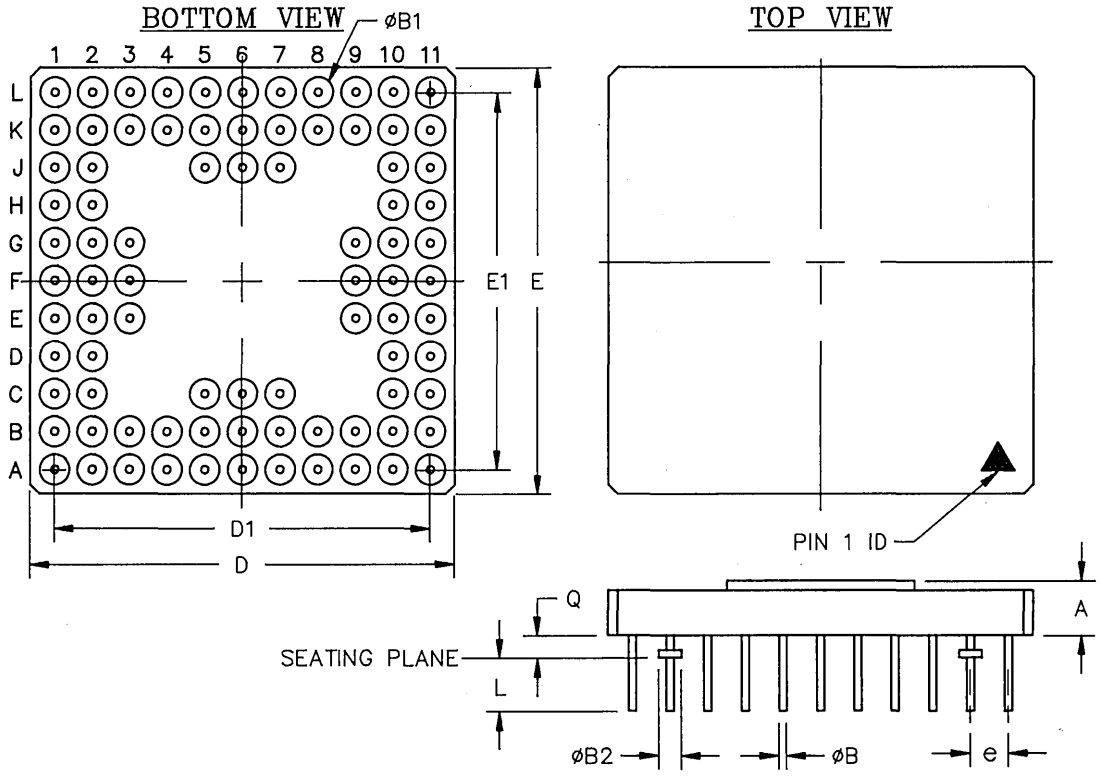


DWG #	G84–1	
# OF PINS (N)	84	
SYMBOL	MIN	MAX
A	.077	.145
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.180	1.235
D1/E1	1.100 BSC	
e	.100 BSC	
L	.120	.140
M	12	
Q	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

84 PIN PGA (CAVITY UP – 11 X 11 GRID)

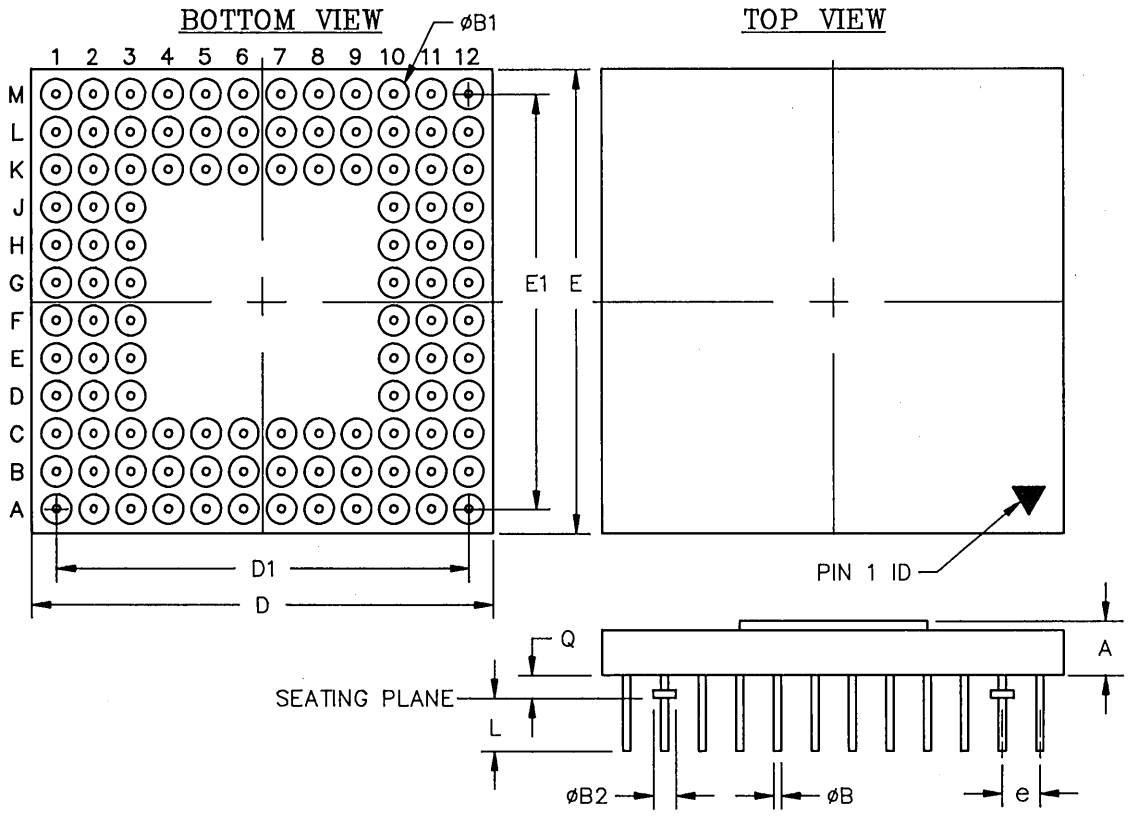


DWG #	G84–3	
# OF PINS (N)	84	
SYMBOL	MIN	MAX
A	.070	.145
ØB	.016	.020
ØB1	–	.080
ØB2	.040	.060
D/E	1.080	1.120
D1/E1	1.000 BSC	
e	.100 BSC	
L	.120	.140
M	11	
Q	.040	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

108 PIN PGA (CAVITY UP)

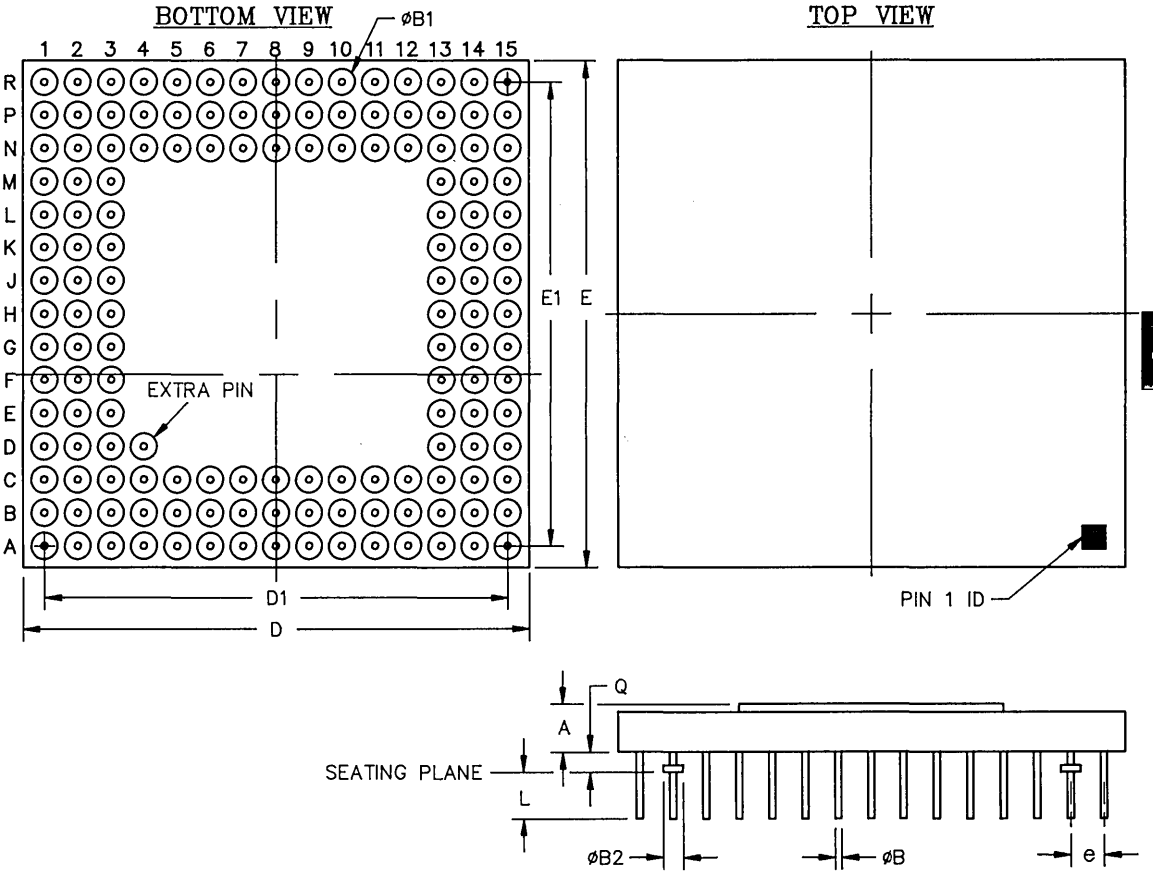


DWG #	G108-1	
# OF PINS (N)	108	
SYMBOL	MIN	MAX
A	.070	.145
ØB	.016	.020
ØB1	–	.080
ØB2	.040	.060
D/E	1.188	1.212
D1/E1	1.100 BSC	
e	.100 BSC	
L	.120	.140
M	12	
Q	.040	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

144 PIN PGA (CAVITY UP – R3001)

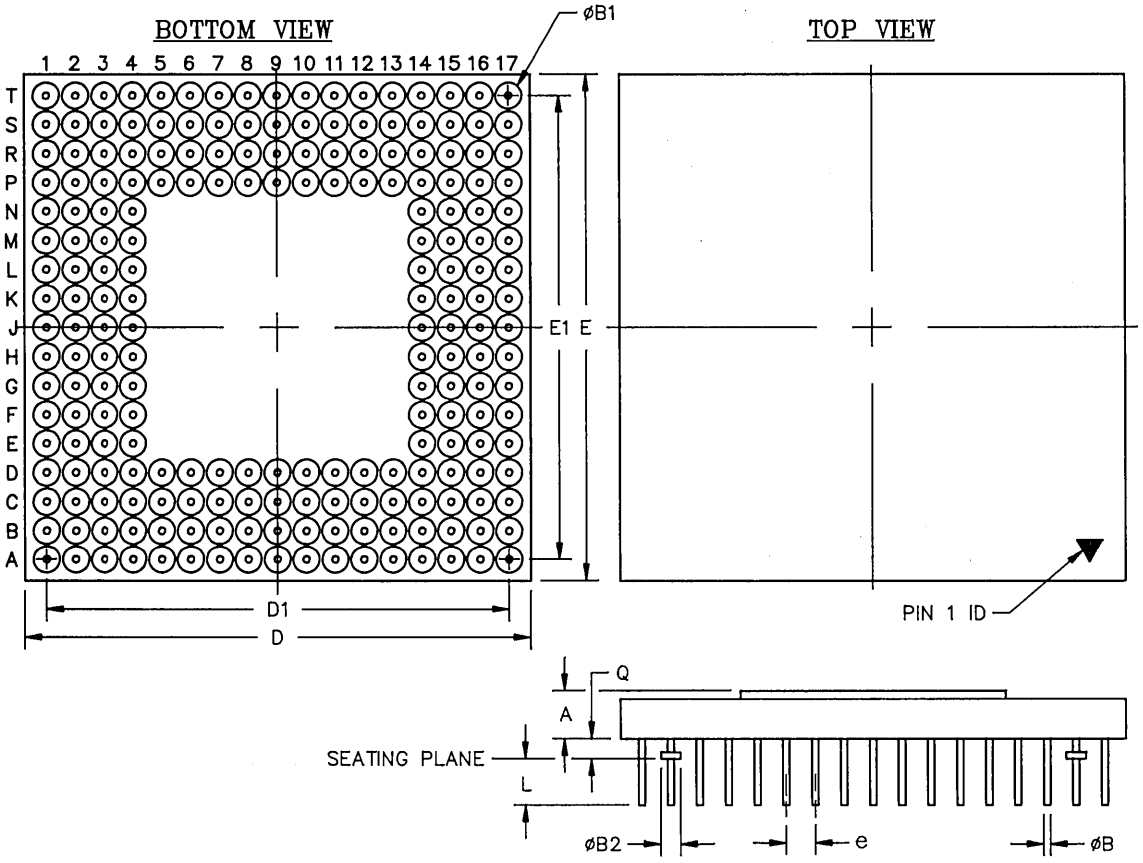


DWG #	G144–2	
# OF PINS (N)	145	
SYMBOL	MIN	MAX
A	.082	.125
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.559	1.590
D1/E1	1.400 BSC	
e	.100 BSC	
L	.120	.140
M	15	
Q	.040	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.
6. EXTRA PIN (D–4) ELECTRICALLY CONNECTED TO D–3.

PIN GRID ARRAYS (Continued)

208 PIN PGA (CAVITY UP)

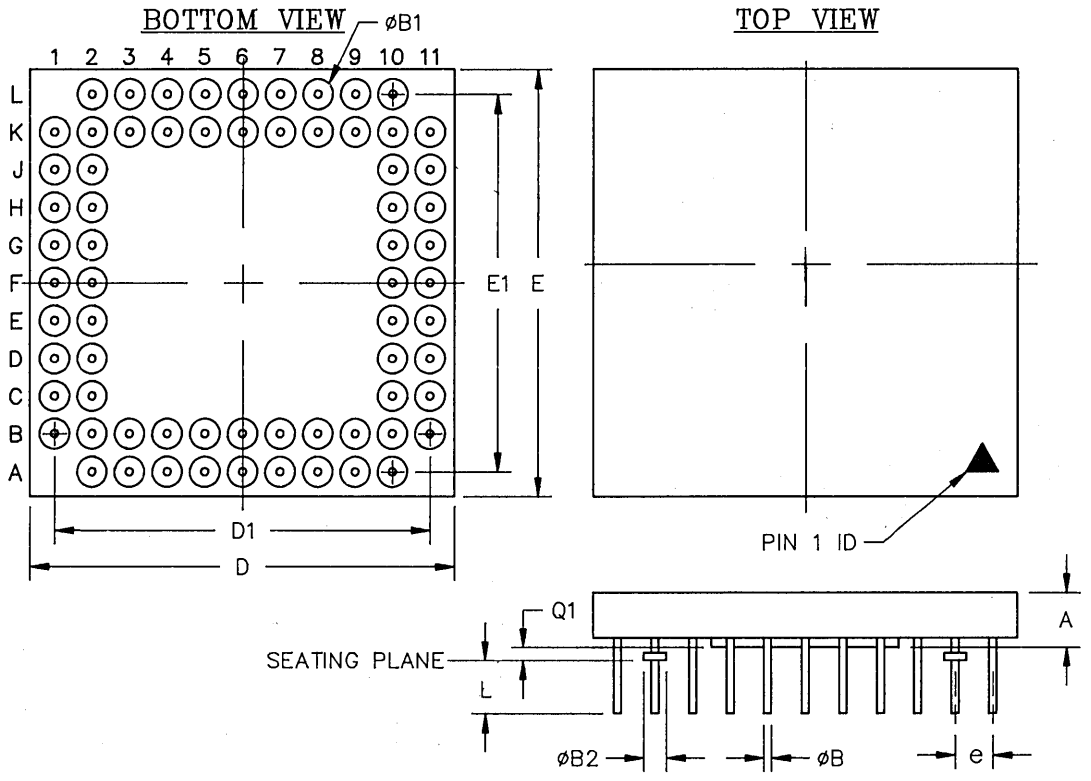


DWG #	G208-1	
# OF PINS (N)	208	
SYMBOL	MIN	MAX
A	.070	.145
øB	.016	.020
øB1	–	.080
øB2	.040	.060
D/E	1.732	1.780
D1/E1	1.600 BSC	
e	.100 BSC	
L	.125	.140
M	17	
Q	.040	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

68 PIN PGA (CAVITY DOWN)

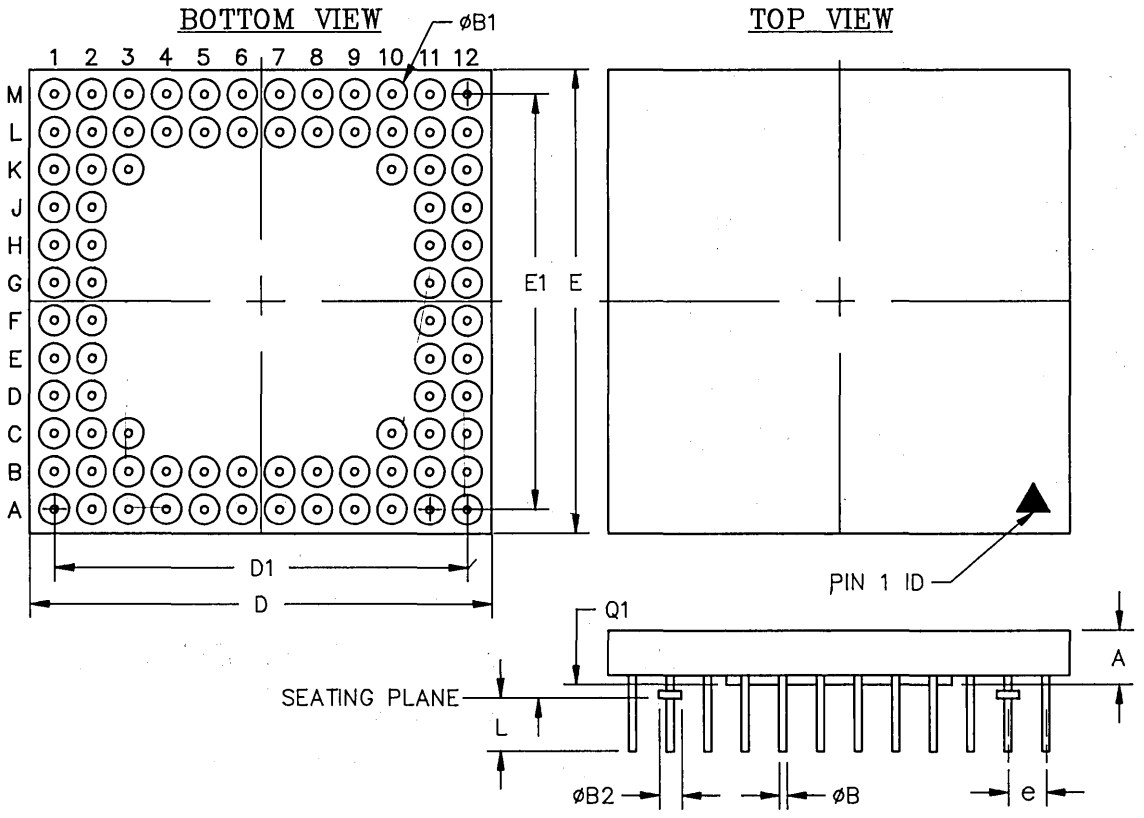


DWG #	GU68-2	
# OF PINS (N)	68	
SYMBOL	MIN	MAX
A	.077	.095
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.098	1.122
D1/E1	1.000 BSC	
e	.100 BSC	
L	.120	.140
M	11	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

84 PIN PGA (CAVITY DOWN)

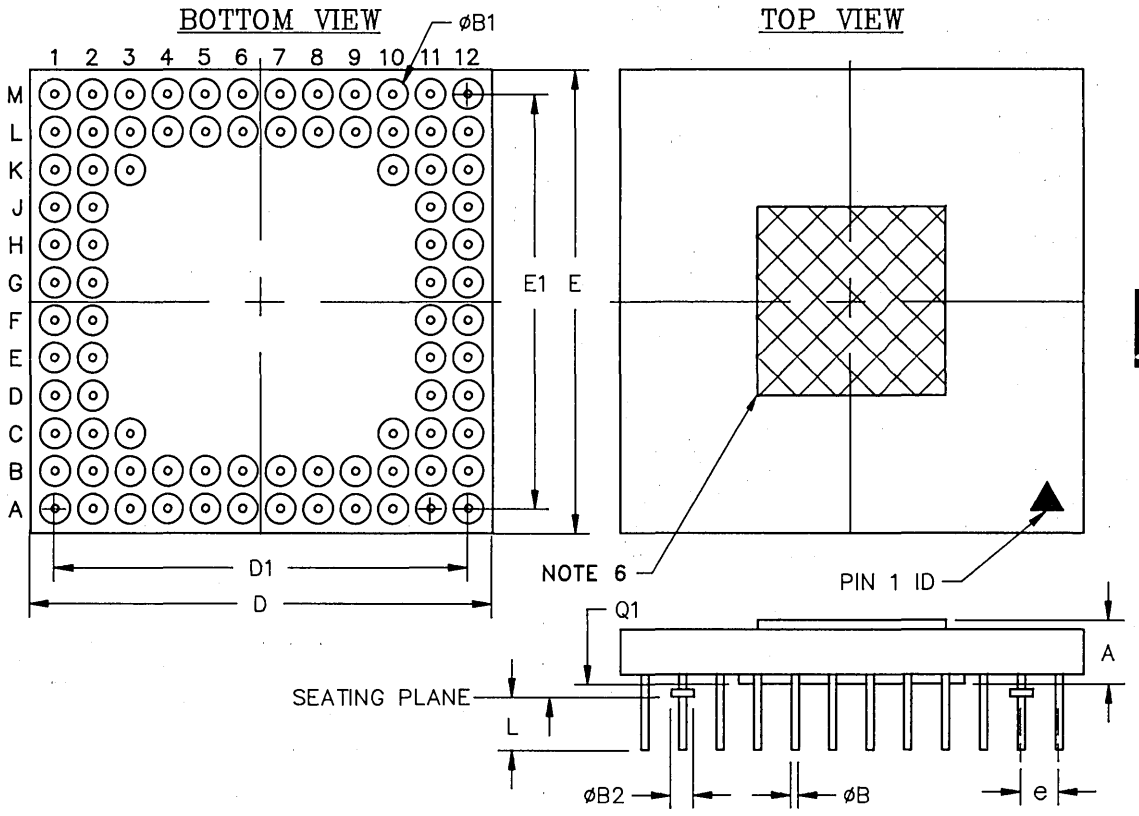


DWG #	G84-2	
# OF PINS (N)	84	
SYMBOL	MIN	MAX
A	.077	.145
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.180	1.235
D1/E1	1.100 BSC	
e	.100 BSC	
L	.120	.140
M	12	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

84 PIN PGA (CAVITY DOWN – R3010A)

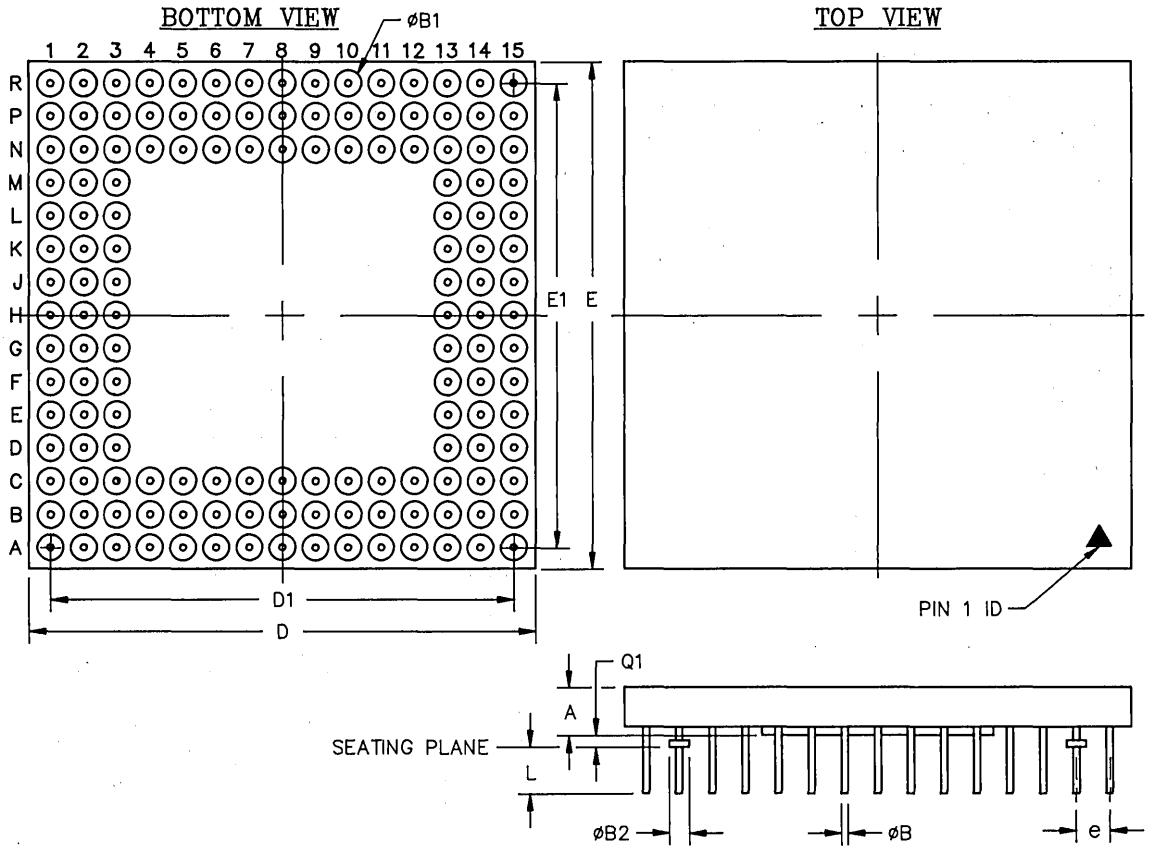


DWG #	G84–4	
# OF PINS (N)	84	
SYMBOL	MIN	MAX
A	.077	.145
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.180	1.235
D1/E1	1.100 BSC	
e	.100 BSC	
L	.120	.140
M	12	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.
6. CROSS HATCHED AREA INDICATES INTEGRALL METALLIC HEAT SINK.

PIN GRID ARRAYS (Continued)

144 PIN PGA (CAVITY DOWN)

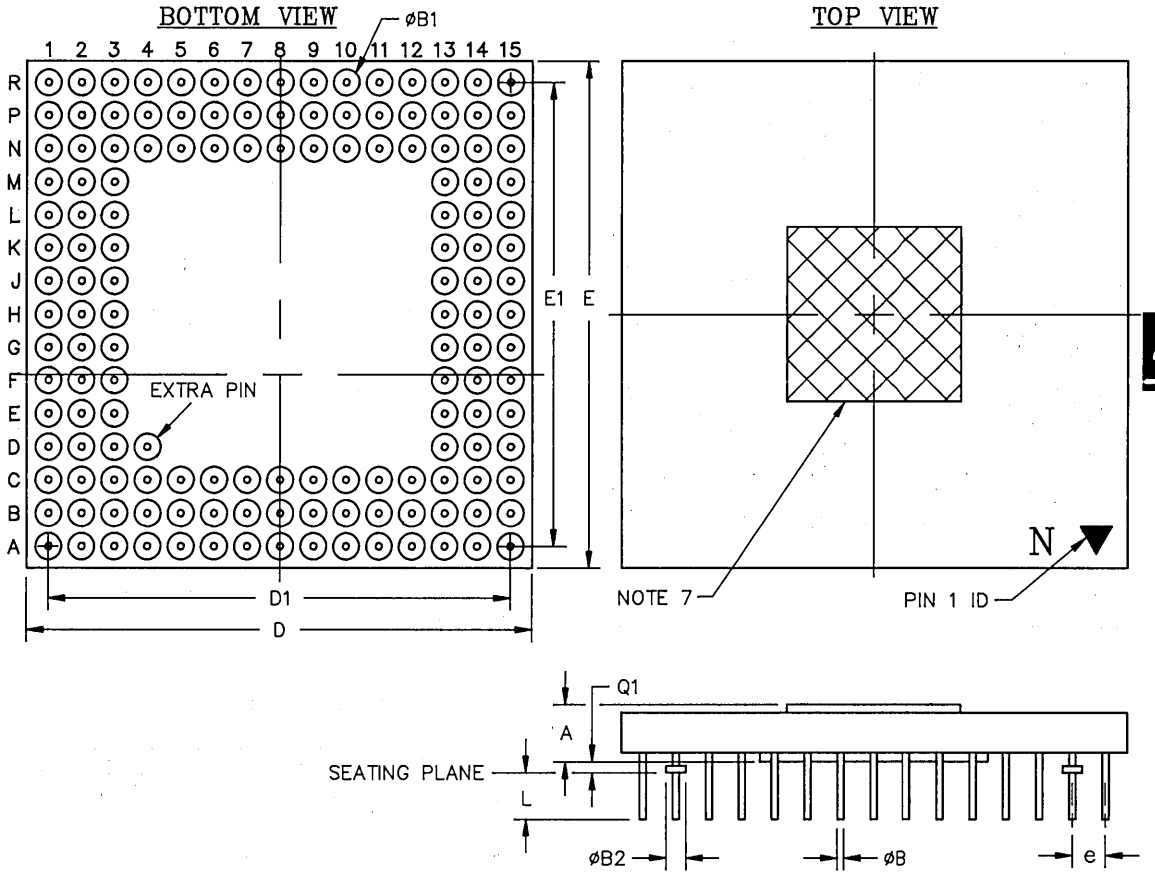


DWG #	G144-1	
# OF PINS (N)	144	
SYMBOL	MIN	MAX
A	.082	.100
ØB	.016	.020
ØB1	.060	.080
ØB2	.040	.060
D/E	1.559	1.590
D1/E1	1.400 BSC	
e	.100 BSC	
L	.120	.140
M	15	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PIN GRID ARRAYS (Continued)

144 PIN PGA (CAVITY DOWN – R3000A)

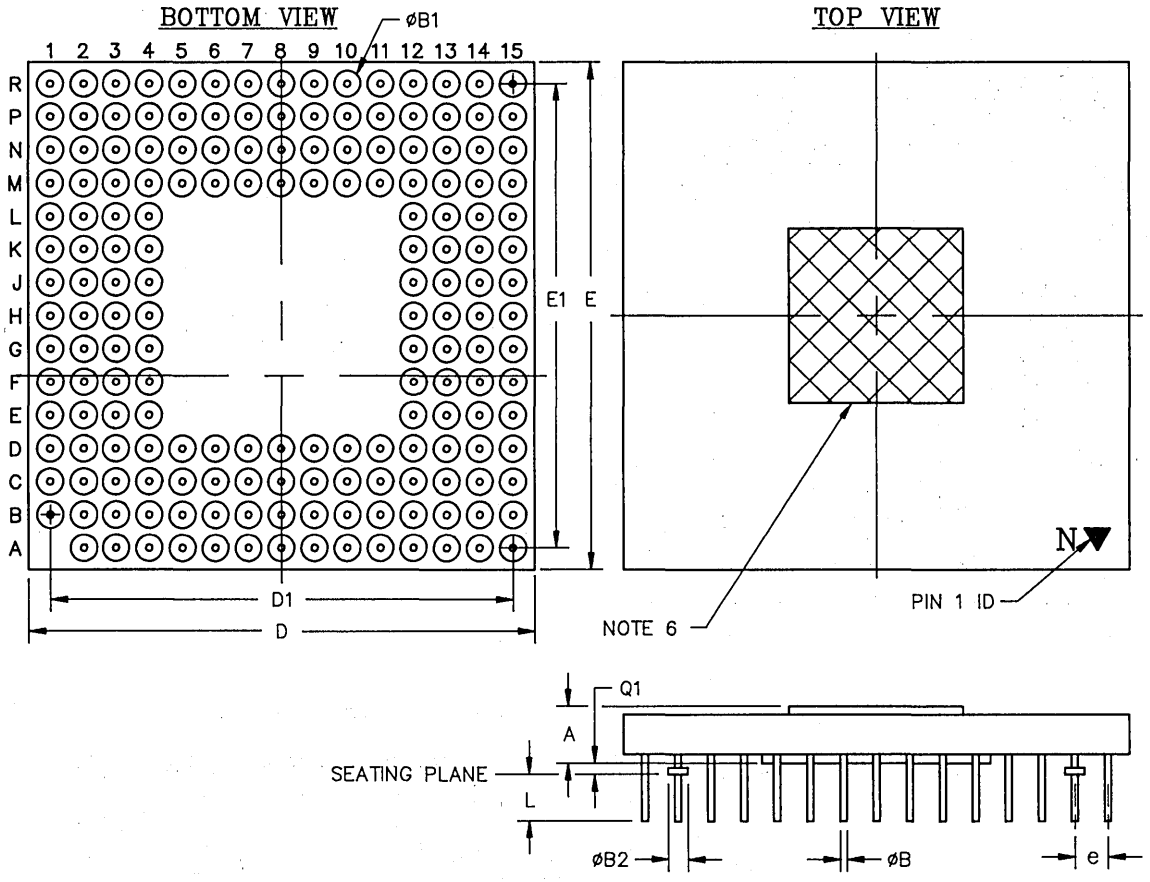


DWG #	G144–3	
# OF PINS (N)	145	
SYMBOL	MIN	MAX
A	.082	.130
ØB	.016	.020
ØB1	.060	.080
ØB2	.040	.060
D/E	1.559	1.590
D1/E1	1.400 BSC	
e	.100 BSC	
L	.120	.140
M	15	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.
6. EXTRA PIN (D–4) ELECTRICALLY CONNECTED TO D–3.
7. CROSS HATCHED AREA INDICATES INTEGRAL METALLIC HEAT SINK.

PIN GRID ARRAYS (Continued)

175 PIN PGA (CAVITY DOWN – R3000A)

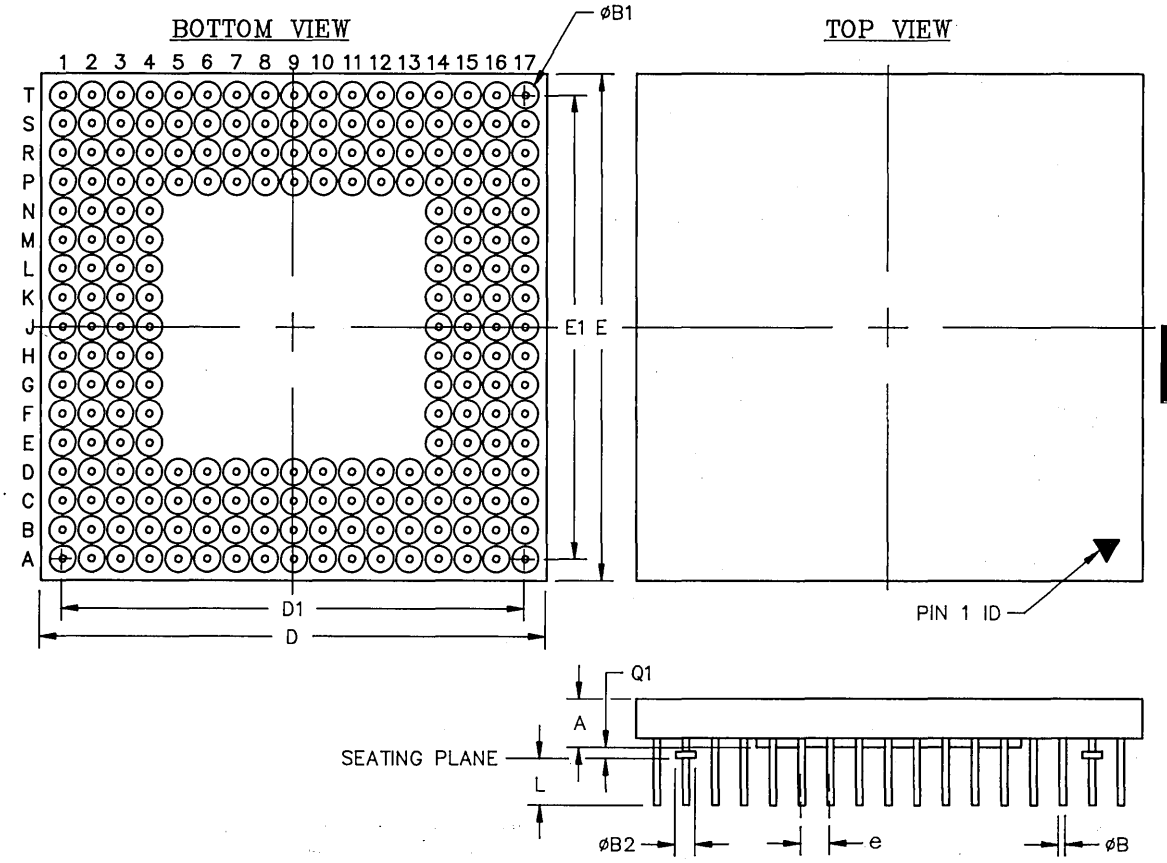


DWG #	G175–1	
# OF PINS (N)	175	
SYMBOL	MIN	MAX
A	.082	.130
øB	.016	.020
øB1	.060	.080
øB2	.040	.060
D/E	1.559	1.590
D1/E1	1.400 BSC	
e	.100 BSC	
L	.120	.140
M	15	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.
6. CROSS HATCHEDAREA INDICATES INTEGRAL METALLIC HEAT SINK.

PIN GRID ARRAYS (Continued)

208 PIN PGA (CAVITY DOWN)

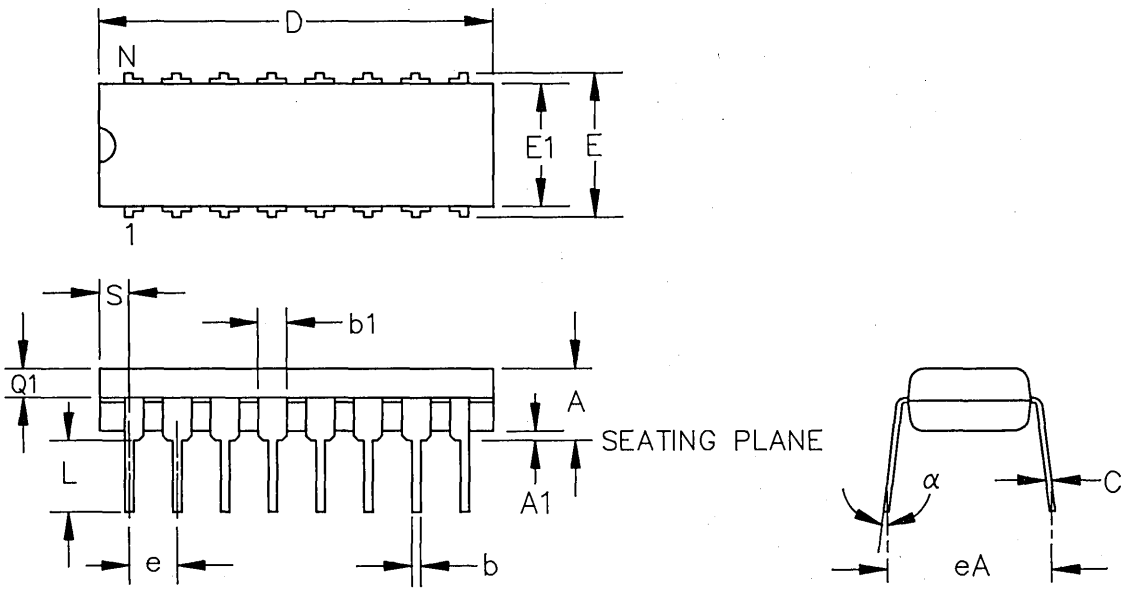


DWG #	G208-2	
# OF PINS (N)	208	
SYMBOL	MIN	MAX
A	.070	.145
øB	.016	.020
øB1	–	.080
øB2	.040	.060
D/E	1.732	1.780
D1/E1	1.600 BSC	
e	.100 BSC	
L	.125	.140
M	17	
Q1	.025	.060

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS
5. CHAMFERED CORNERS ARE IDT'S OPTION.

PLASTIC DUAL IN-LINE PACKAGES

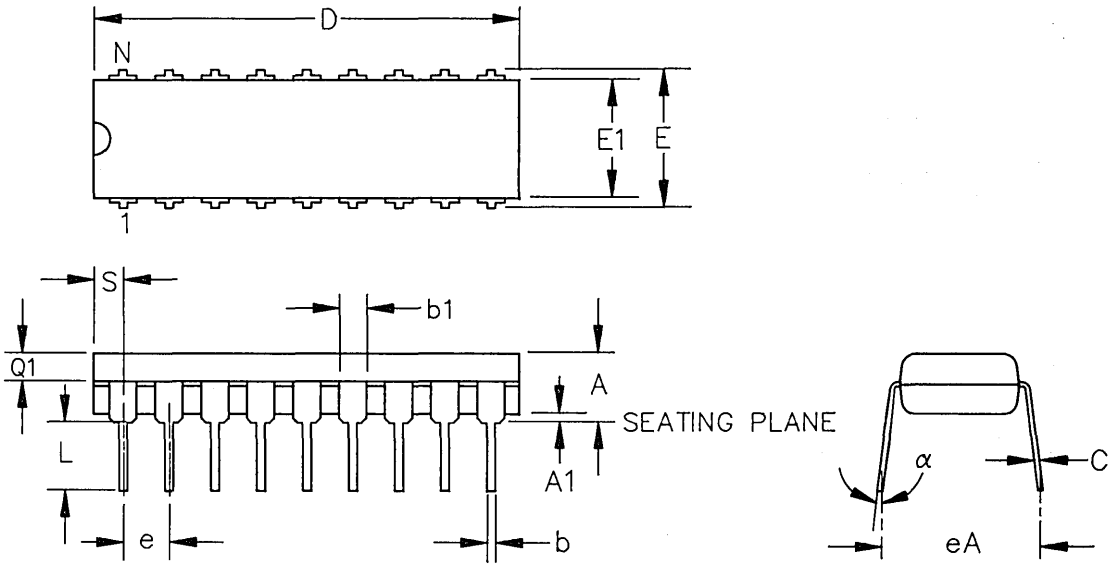
16-32 LEAD PLASTIC DIP (300 MIL)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. D & E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

DWG #	P16-1		P22-1		P28-2		P32-2	
# OF LDS (N)	16		22		28		32	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.140	.165	.145	.165	.145	.180	.145	.180
A1	.015	.035	.015	.035	.015	.030	.015	.030
b	.015	.022	.015	.022	.015	.022	.016	.022
b1	.050	.070	.050	.065	.045	.065	.045	.060
C	.008	.012	.008	.012	.008	.015	.008	.015
D	.745	.760	1.050	1.060	1.345	1.375	1.545	1.585
E	.300	.325	.300	.320	.300	.325	.300	.325
E1	.247	.260	.240	.270	.270	.295	.275	.295
e	.090	.110	.090	.110	.090	.110	.090	.110
eA	.310	.370	.310	.370	.310	.400	.310	.400
L	.120	.150	.120	.150	.120	.150	.120	.150
α	0°	15°	0°	15°	0°	15°	0°	15°
S	.015	.035	.020	.040	.020	.042	.020	.060
Q1	.050	.070	.055	.075	.055	.065	.055	.065

PLASTIC DUAL IN-LINE PACKAGES (Continued)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. D & E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

18-24 LEAD PLASTIC DIP (300 MIL - FULL LEAD)

DWG #	P18-1		P20-1		P24-1	
# OF LDS (N)	18		20		24	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX
A	.140	.165	.145	.165	.145	.165
A1	.015	.035	.015	.035	.015	.035
b	.015	.020	.015	.020	.015	.020
b1	.050	.070	.050	.070	.050	.065
C	.008	.012	.008	.012	.008	.012
D	.885	.910	1.022	1.040	1.240	1.255
E	.300	.325	.300	.325	.300	.320
E1	.247	.260	.240	.280	.250	.275
e	.090	.110	.090	.110	.090	.110
eA	.310	.370	.310	.370	.310	.370
L	.120	.150	.120	.150	.120	.150
α	0°	15°	0°	15°	0°	15°
S	.040	.060	.025	.070	.055	.075
Q1	.050	.070	.055	.075	.055	.070

PLASTIC DUAL IN-LINE PACKAGES (Continued)

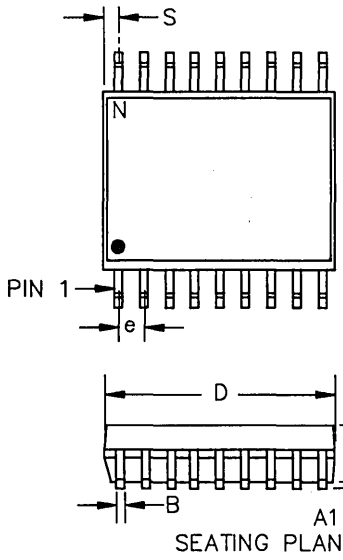
24-48 LEAD PLASTIC DIP (600 MIL)

DWG #	P24-2		P28-1		P32-1		P40-1		P48-1	
# OF LEADS (N)	24		28		32		40		48	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.160	.185	.160	.185	.170	.190	.160	.185	.170	.200
A1	.015	.035	.015	.035	.015	.050	.015	.035	.015	.035
b	.015	.020	.015	.020	.016	.020	.015	.020	.015	.020
b1	.050	.065	.050	.065	.045	.055	.050	.065	.050	.065
C	.008	.012	.008	.012	.008	.012	.008	.012	.008	.012
D	1.240	1.260	1.420	1.460	1.645	1.655	2.050	2.070	2.420	2.450
E	.600	.620	.600	.620	.600	.625	.600	.620	.600	.620
E1	.530	.550	.530	.550	.530	.550	.530	.550	.530	.560
e	.090	.110	.090	.110	.090	.110	.090	.110	.090	.110
eA	.610	.670	.610	.670	.610	.670	.610	.670	.610	.670
L	.120	.150	.120	.150	.125	.135	.120	.150	.120	.150
α	0°	15°	0°	15°	0°	15°	0°	15°	0°	15°
S	.060	.080	.055	.080	.070	.080	.070	.085	.060	.075
Q1	.060	.080	.060	.080	.065	.075	.060	.080	.060	.080

64 LEAD PLASTIC DIP (900 MIL)

DWG #	P64-1	
# OF LEADS (N)	64	
SYMBOLS	MIN	MAX
A	.180	.230
A1	.015	.040
b	.015	.020
b1	.050	.065
C	.008	.012
D	3.200	3.220
E	.900	.925
E1	.790	.810
e	.090	.110
eA	.910	1.000
L	.120	.150
α	0°	15°
S	.045	.065
Q1	.080	.090

SMALL OUTLINE IC



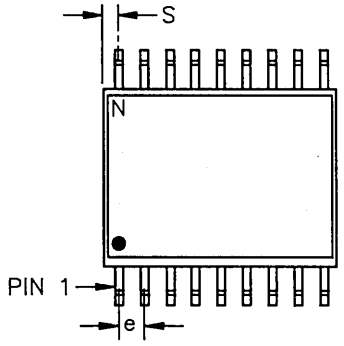
NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D & E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS AND TO BE MEASURED FROM THE BOTTOM OF PKG.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.

4

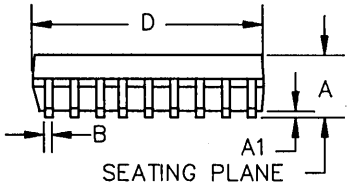
16–24 LEAD SMALL OUTLINE (GULL WING)

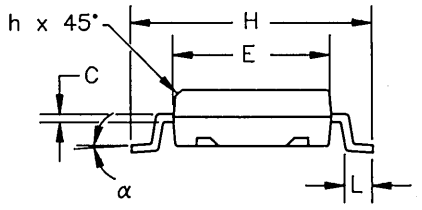
DWG #	SO16–1		SO18–1		SO20–2		SO24–2		SO24–3	
# OF LDS (N)	16 (.300)		18 (.300)		20 (.300")		24 (.300")		24 (.300")	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.095	.1043	.095	.1043	.095	.1043	.095	.1043	.110	.120
A1	.005	.0118	.005	.0118	.005	.0118	.005	.0118	.005	.0118
B	.014	.020	.014	.020	.014	.020	.014	.020	.014	.020
C	.0091	.0125	.0091	.0125	.0091	.0125	.0091	.0125	.007	.011
D	.403	.413	.447	.462	.497	.511	.600	.614	.620	.630
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
E	.292	.2992	.292	.2992	.292	.2992	.292	.2992	.295	.305
h	.010	.020	.010	.020	.010	.020	.010	.020	.012	.020
H	.400	.419	.400	.419	.400	.419	.400	.419	.406	.419
L	.018	.045	.018	.045	.018	.045	.018	.045	.028	.045
α	0°	8°	0°	8°	0°	8°	0°	8°	0°	8°
S	.023	.035	.023	.035	.023	.035	.023	.035	.032	.043

SMALL OUTLINE IC (Continued)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D & E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS AND TO BE MEASURED FROM THE BOTTOM OF THE PKG.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.





28 LEAD SMALL OUTLINE (GULL WING)

DWG #	SO28–2		SO28–3	
# OF LDS (N)	28 (.300")		28 (.330")	
SYMBOL	MIN	MAX	MIN	MAX
A	.095	.1043	.110	.120
A1	.005	.0118	.005	.014
B	.014	.020	.014	.019
C	.0091	.0125	.006	.010
D	.700	.712	.718	.728
e	.050 BSC		.050 BSC	
E	.292	.2992	.340	.350
h	.010	.020	.012	.020
H	.400	.419	.462	.478
L	.018	.045	.028	.045
α	0°	8°	0°	8°
S	.023	.035	.023	.035

SMALL OUTLINE IC (Continued)

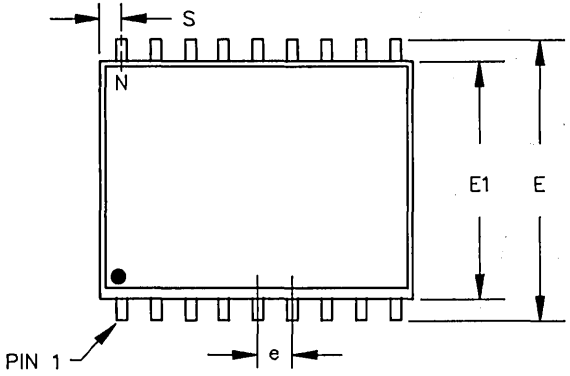
16–24 LEAD SMALL OUTLINE (EIAJ – .0315 PITCH)

DWG #	SO16–5		SO20–5		SO24–5	
# OF LDS (N)	16		20		24	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX
A	.057	.071	.069	.083	.069	.083
A1	.002 TYP		.002 TYP		.002 TYP	
B	.012	.020	.012	.020	.012	.020
C	.006	.010	.006	.010	.006	.010
D	.248	.271	.331	.354	.382	.405
E	.165	.180	.205	.220	.205	.220
e	.0315 BSC		.0315 BSC		.0315 BSC	
H	.232	.256	.295	.319	.295	.319
L	.010	–	.010	–	.010	–
α	0°	8°	0°	8°	0°	8°

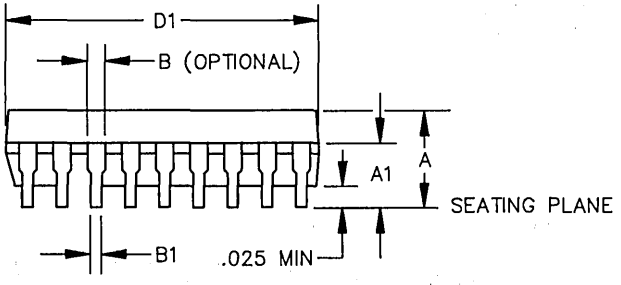
16–28 LEAD SMALL OUTLINE (EIAJ – .050 PITCH)

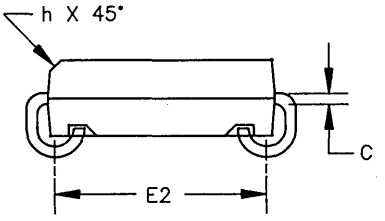
DWG #	SO16–6		SO18–6		SO20–6		SO24–6		SO28–6	
# OF LDS (N)	16		18		20		24		28	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.057	.071	.069	.083	.069	.083	.069	.083	.083	.098
A1	.002 TYP		.002 TYP		.002 TYP		.002 TYP		.002 TYP	
B	.012	.020	.012	.020	.012	.020	.012	.020	.012	.020
C	.006	.010	.006	.010	.006	.010	.006	.010	.006	.010
D	.382	.406	.437	.453	.480	.504	.580	.603	.720	.740
E	.165	.180	.205	.220	.205	.220	.205	.220	.290	.300
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
H	.232	.256	.295	.319	.295	.319	.295	.319	.378	.402
L	.010	–	.010	–	.010	–	.010	–	.010	–
α	0°	8°	0°	8°	0°	8°	0°	8°	0°	8°

SMALL OUTLINE IC (Continued)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D1 & E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSION AND TO BE MEASURED FROM THE BOTTOM OF THE PKG.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE

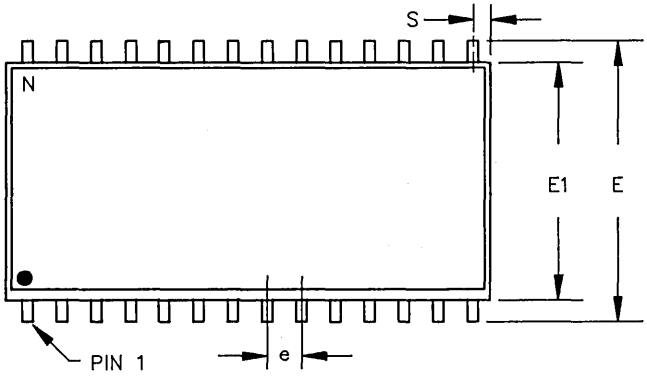




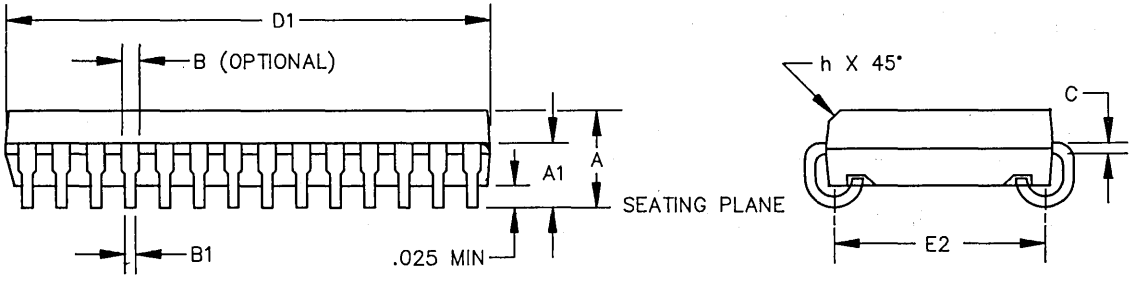
16–24 LEAD SMALL OUTLINE (J–BEND)

DWG #	SO16–2		SO20–1		SO24–4	
# OF LDS (N)	16 LD (.300")		20 LD (.300")		24 LD (.300")	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX
A	.120	.140	.120	.140	.130	.148
A1	.078	.095	.078	.095	.082	.095
B	.020	.024	.020	.024	.026	.032
B1	.014	.020	.014	.020	.015	.020
C	.008	.013	.008	.013	.007	.011
D1	.400	.412	.500	.512	.620	.630
E	.335	.347	.335	.347	.335	.345
E1	.292	.300	.292	.300	.295	.305
E2	.262	.272	.262	.272	.260	.280
e	.050 BSC		.050 BSC		.050 BSC	
h	.010	.020	.010	.020	.010	.020
S	.023	.035	.023	.035	.032	.043

SMALL OUTLINE IC (Continued)



NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D1 & E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSION AND TO BE MEASURED FROM THE BOTTOM OF THE PKG.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.

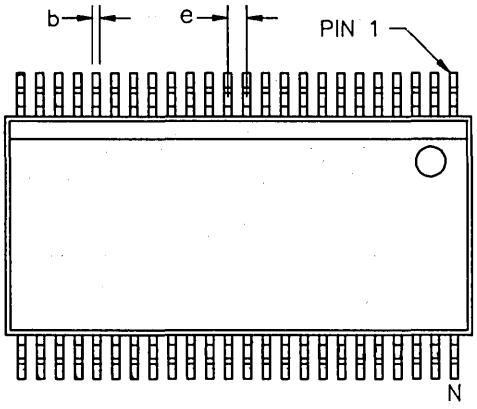


28–32 LEAD SMALL OUTLINE (J–BEND)

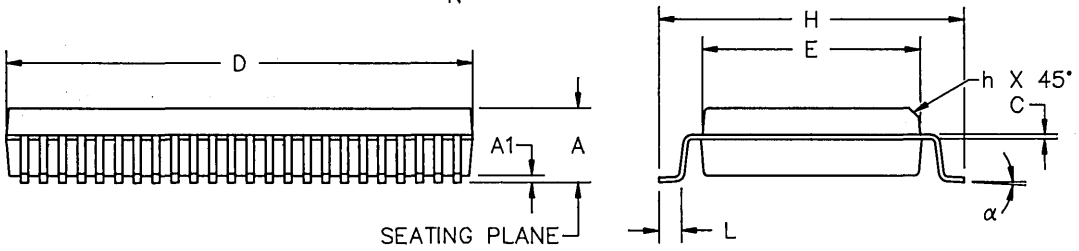
DWG #	SO28–5		SO28–4		SO32–2	
# OF LDS (N)	28 LD (.300")		28 LD (.350")		32 LD (.300")	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX
A	.120	.140	.130	.148	.130	.148
A1	.078	.095	.082	.095	.082	.095
B	.020	.024	.026	.032	.026	.032
B1	.014	.020	.016	.020	.016	.020
C	.008	.013	.007	.011	.008	.013
D1	.700	.712	.720	.730	.820	.830
E	.335	.347	.380	.390	.330	.340
E1	.292	.300	.345	.355	.295	.305
E2	.262	.272	.310	.330	.260	.275
e	.050 BSC		.050 BSC		.050 BSC	
h	.012	.020	.012	.020	.012	.020
S	.023	.035	.023	.035	.032	.043

SMALL OUTLINE IC (Continued)

48 & 56 LEAD SMALL OUTLINE (SSOP – GULL WING)



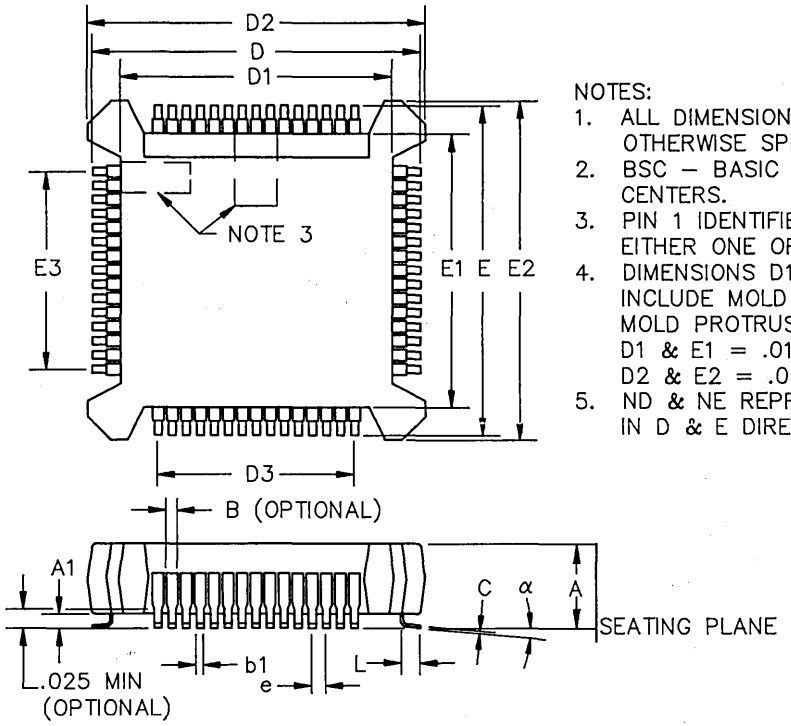
NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D & E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.



DWG #	SO48–1		SO56–1	
# OF LDS (N)	48 (.300")		56 (.300")	
SYMBOL	MIN	MAX	MIN	MAX
A	.095	.110	.095	.110
A1	.008	.016	.008	.016
b	.008	.012	.008	.012
C	.005	.009	.005	.009
D	.620	.630	.720	.730
E	.291	.299	.291	.299
e	.025 BSC		.025 BSC	
H	.395	.420	.395	.420
h	.015	.025	.015	.025
L	.020	.040	.020	.040
α	0°	8°	0°	8°

PLASTIC QUAD FLATPACKS

100–132 LEAD PLASTIC QUAD FLATPACK (JEDEC)

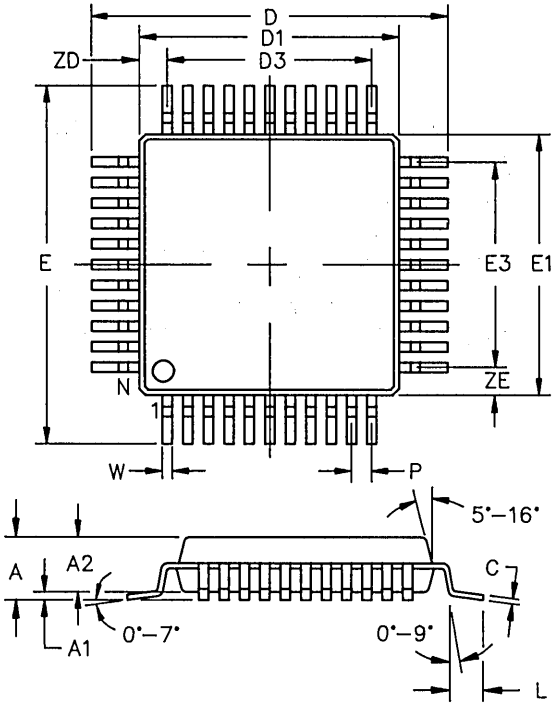


NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. PIN 1 IDENTIFIER CAN BE POSITIONED AT EITHER ONE OF THESE TWO LOCATIONS.
4. DIMENSIONS D1, D2, E1, AND E2 DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE MOLD PROTRUSIONS ARE AS FOLLOWS:
   D1 & E1 = .010 MAX.
   D2 & E2 = .007 MAX.
5. ND & NE REPRESENT NUMBERS OF LEADS IN D & E DIRECTIONS RESPECTIVELY.

DWG #	PQ100–1		PQ132–1	
# OF LDS (N)	100		132	
SYMBOLS	MIN	MAX	MIN	MAX
A	.160	.180	.160	.180
A1	.020	.040	.020	.040
B	.008	.016	.008	.016
b1	.008	.012	.008	.012
C	.0055	.008	.0055	.008
D	.875	.885	1.075	1.085
D1	.747	.753	.947	.953
D2	.897	.903	1.097	1.103
D3	.600 REF		.800 REF	
e	.025 BSC		.025 BSC	
E	.875	.885	1.075	1.085
E1	.747	.753	.947	.953
E2	.897	.903	1.097	1.103
E3	.600 REF		.800 REF	
L	.020	.030	.020	.030
α	0°	8°	0°	8°
ND/NE	25/25		33/33	

PLASTIC QUAD FLATPACKS (Continued)

80–128 LEAD PLASTIC QUAD FLATPACK (EIAJ)



NOTES:

1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D1 & E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS .010 PER SIDE.
4. ND & NE REPRESENT NUMBERS OF LEADS IN D & E DIRECTIONS RESPECTIVELY.

[5] THE 3.9mm FOOTPRINT IS STANDARD, HOWEVER THE 3.2mm IS OPTIONAL & CAN BE REQUESTED.

DWG #	PQ80–2		PQ100–2		PQ120–2		PQ128–2	
# OF LDS (N)	80		100		120		128	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.110	.124	.110	.124	.136	.156	.136	.156
A1	.010	–	.010	–	.010	–	.010	–
A2	.100	.120	.100	.120	.125	.144	.125	.144
C	.005	.008	.005	.008	.005	.008	.005	.008
D	.937	.945	.937	.945	1.252	1.260	1.252	1.260
D1	.783	.791	.783	.791	1.098	1.106	1.098	1.106
D3	.724 REF		.742 REF		.913 REF		.976 REF	
E	.701	.709	.701	.709	1.252	1.260	1.252	1.260
E1	.547	.555	.547	.555	1.098	1.106	1.098	1.106
E3	.472 REF		.486 REF		.913 REF		.976 REF	
L	.026	.037	.026	.037	.026	.037	.026	.037
ND/NE	16/24		20/30		30/30		32/32	
P	.0315 BSC		.026 BSC		.026 BSC		.0315 BSC	
W	.012	.018	.012	.018	.012	.018	.012	.018
ZD	.032		.023		.094		.063	
ZE	.039		.032		.094		.063	

ALT. D [5]	.909	.917	.909	.917	1.224	1.232	1.224	1.232
ALT. E [5]	.673	.681	.673	.681	1.224	1.232	1.224	1.232

PLASTIC QUAD FLATPACKS (Continued)

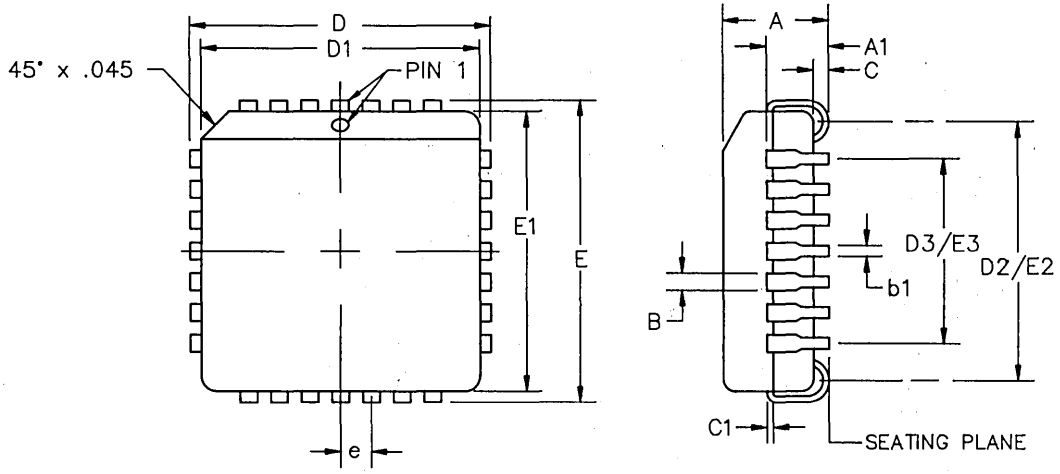
144–208 LEAD PLASTIC QUAD FLATPACK (EIAJ)

DWG #	PQ144–2		PQ160–2		PQ184–2		PQ208–2	
# OF LDS (N)	144		160		184		208	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.136	.156	.136	.156	.136	.156	.136	.156
A1	.010	–	.010	–	.010	–	.010	–
A2	.125	.144	.125	.144	.125	.144	.125	.144
C	.005	.008	.005	.008	.005	.008	.005	.008
D	1.252	1.260	1.252	1.260	1.252	1.260	1.252	1.260
D1	1.098	1.106	1.098	1.106	1.098	1.106	1.098	1.106
D3	.896 REF		.998 REF		.886 REF		1.004 REF	
E	1.252	1.260	1.252	1.260	1.252	1.260	1.252	1.260
E1	1.098	1.106	1.098	1.106	1.098	1.106	1.098	1.106
E3	.896 REF		.998 REF		.886 REF		1.004 REF	
L	.026	.037	.026	.037	.026	.037	.026	.037
ND/NE	36/36		40/40		46/46		52/52	
P	.026 BSC		.026 BSC		.020 BSC		.020 BSC	
W	.009	.014	.009	.014	.009	.014	.009	.014
ZD	.103		.052		.108		.049	
ZE	.103		.052		.108		.049	

ALT. D [5]	1.224	1.232	1.224	1.232	1.224	1.232	1.224	1.232
ALT. E [5]	1.224	1.232	1.224	1.232	1.224	1.232	1.224	1.232

PLASTIC LEADED CHIP CARRIERS

20–84 LEAD PLCC (SQUARE)



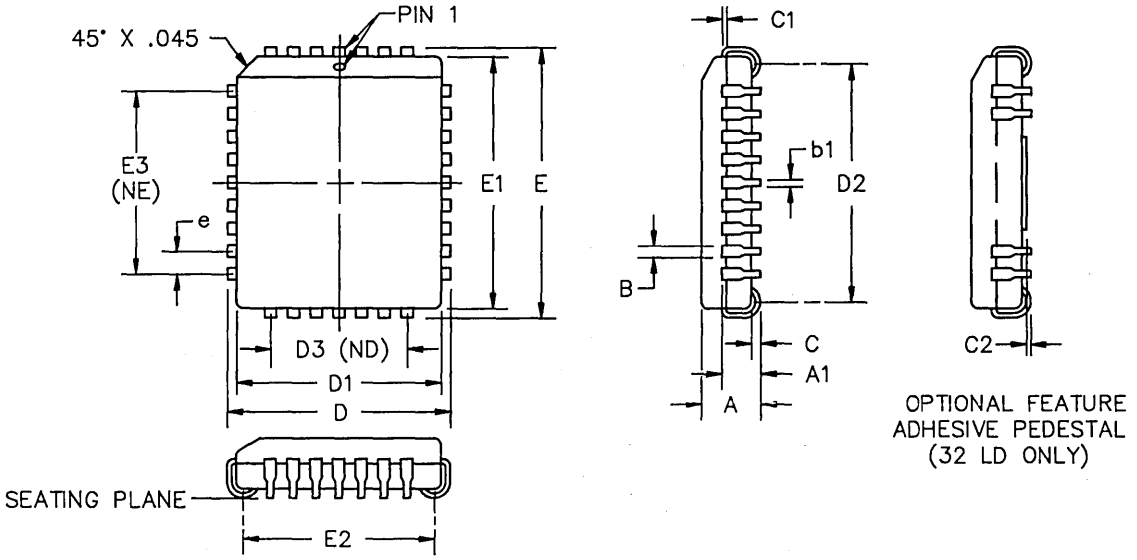
NOTES:

1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS
3. D & E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.
5. ND & NE REPRESENT NUMBER OF LEADS IN D & E DIRECTIONS RESPECTIVELY.
6. D1 & E1 SHOULD BE MEASURED FROM THE BOTTOM OF THE PKG.

DWG #	J20–1		J28–1		J44–1		J52–1		J68–1		J84–1	
# OF LDS	20		28		44		52		68		84	
SYMBOL	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
A	.165	.180	.165	.180	.165	.180	.165	.180	.165	.180	.165	.180
A1	.095	.115	.095	.115	.095	.115	.095	.115	.095	.115	.095	.115
B	.026	.032	.026	.032	.026	.032	.026	.032	.026	.032	.026	.032
b1	.013	.021	.013	.021	.013	.021	.013	.021	.013	.021	.013	.021
C	.020	.040	.020	.040	.020	.040	.020	.040	.020	.040	.020	.040
C1	.008	.012	.008	.012	.008	.012	.008	.012	.008	.012	.008	.012
D	.385	.395	.485	.495	.685	.695	.785	.795	.985	.995	1.185	1.195
D1	.350	.356	.450	.456	.650	.656	.750	.756	.950	.956	1.150	1.156
D2/E2	.290	.330	.390	.430	.590	.630	.690	.730	.890	.930	1.090	1.130
D3/E3	.200 REF		.300 REF		.500 REF		.600 REF		.800 REF		1.000 REF	
E	.385	.395	.485	.495	.685	.695	.785	.795	.985	.995	1.185	1.195
E1	.350	.356	.450	.456	.650	.656	.750	.756	.950	.956	1.150	1.156
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
ND/NE	5		7		11		13		17		21	

PLASTIC LEADED CHIP CARRIERS (Continued)

18–32 LEAD PLCC (RECTANGULAR)

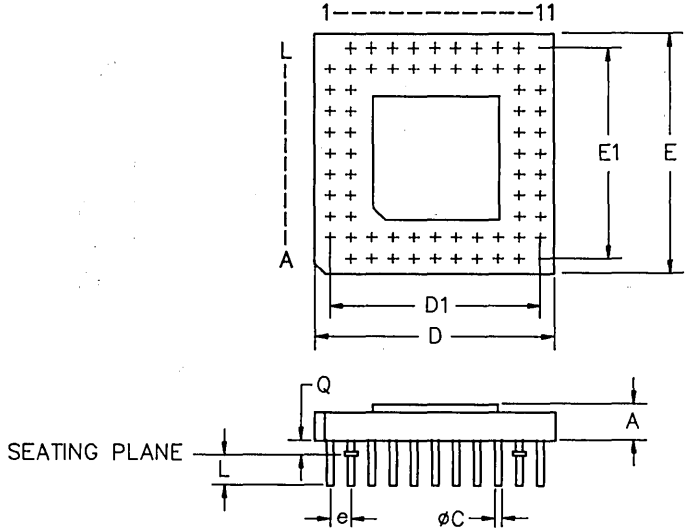


DWG #	J18–1		J32–1	
# OF LDS	18		32	
SYMBOL	MIN	MAX	MIN	MAX
A	.120	.140	.120	.140
A1	.075	.095	.075	.095
B	.026	.032	.026	.032
b1	.013	.021	.013	.021
C	.015	.040	.015	.040
C1	.008	.012	.008	.012
C2	–	–	.005	.015
D	.320	.335	.485	.495
D1	.289	.293	.449	.453
D2	.225	.265	.390	.430
D3	.150 REF		.300 REF	
E	.520	.535	.585	.595
E1	.489	.493	.549	.553
E2	.422	.465	.490	.530
E3	.200 REF		.400 REF	
e	.050 BSC		.050 BSC	
ND/NE	4 / 5		7 / 9	

NOTES:
1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC LEAD SPACING BETWEEN CENTERS.
3. D & E DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.
4. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN .004" AT THE SEATING PLANE.
5. ND & NE REPRESENT NUMBERS OF LEADS IN D & E DIRECTIONS RESPECTIVELY.
6. D1 & E1 SHOULD BE MEASURED FROM THE BOTTOM OF THE PACKAGE.

PLASTIC PIN GRID ARRAYS

68–208 PIN PGA (CAVITY UP)



NOTES:

1. ALL DIMENSIONS ARE IN INCHES, UNLESS OTHERWISE SPECIFIED.
2. BSC – BASIC PIN SPACING BETWEEN CENTERS.
3. SYMBOL "M" REPRESENTS THE PGA MATRIX SIZE.
4. SYMBOL "N" REPRESENTS THE NUMBER OF PINS.
5. DIM. "A" INCLUDES BOTH THE PKG BODY & THE LID. IT DOES NOT INCLUDE HEATSINK OR OTHER ATTACHED FEATURES.
6. PIN DIAMETER "C" EXCLUDES SOLDER DIP OR OTHER LEAD FINISH.
7. PIN TIPS MAY HAVE RADIUS OR CHAMFER.

DWG No.	PG 68–2		PG 84–2		PG 208–2	
# OF PINS (N)	68 PIN		84 PIN		208 PIN	
SYMBOLS	MIN	MAX	MIN	MAX	MIN	MAX
A	.115	.160	.115	.160	.115	.160
C	.016	.020	.016	.020	.016	.020
D	1.140	1.180	1.140	1.180	1.740	1.780
D1	1.000 BSC		1.000 BSC		1.600 BSC	
E	1.140	1.180	1.140	1.180	1.740	1.780
E1	1.000 BSC		1.000 BSC		1.600 BSC	
e	.100 BSC		.100 BSC		.100 BSC	
L	.100	.160	.100	.160	.100	.160
M	11		11		17	
Q	.040	.070	.040	.070	.040	.070

IDT RISC PROCESSING COMPONENTS

THE COMPLETE RISC SOLUTION

Integrated Device Technology, Inc. is dedicated to providing complete RISC design solutions by combining expertise in silicon processes with leadership products in development systems and software. Long an industry leader in the fastest static RAMs and high-speed logic, IDT now offers RISC system building blocks comprised of components and board-level subsystems.

As a semiconductor partner with MIPS Computer Systems, IDT has established a leadership position in the RISC marketplace by supplying the fastest CPUs at 40MHz, pioneering RISC CPU Subsystem™ modules, and offering cost-effective development tools and software.

The MIPS architecture has become an industry standard and has been adopted by over 100 leading OEM manufacturers including DEC, Sony, Tandem, NEC, CDC, Adobe, Siemens, Nixdorf, Honeywell Bull and Silicon Graphics. The MIPS ISA (Instruction Set Architecture) has been selected by JIAWG as the 32-bit microprocessor standard for military avionics.

RISComponent™ FAMILY OVERVIEW

The R3000 Family consists of the R3000 RISC CPU, the R3001 and R3051/52 RISControllers™, the R3010 Floating-Point Accelerator, the R3500 RISCore™ and the R3020 Write Buffer. The R3000 processor is a derivative of the R2000A, the first commercially available RISC processor introduced in 1985. The R3001 RISController and the R3051 family are versions of the processor tailored for embedded control and low-cost workstations. The R3500 integrates floating point capability onto the R3000 pinout. The R4000 is the third generation of the MIPS RISC architecture that sets a new performance standard for the 1990s.

THE IDT79R3000 CPU

The R3000 processor consists of two tightly-coupled processors implemented on a single chip.

The first processor is a full 32-bit Harvard Architecture CPU consisting of 32 registers, an integer ALU, a single-cycle shifter and a multiplier/divider. The second processor is a system control coprocessor containing a Translation Look-aside Buffer (TLB) and control registers to support a virtual memory space of 4GBytes and separate Instruction and Data caches.

The R3000 CPU features:

- Full 32-bit operation
- Three instruction formats
- Efficient 5-stage pipeline
- On-chip cache control
- On-chip Memory Management Unit
- Multiprocessor capability

5

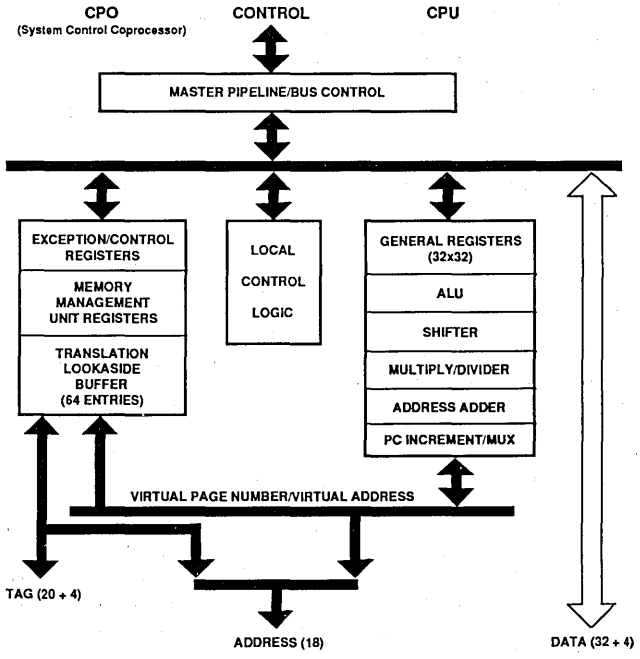


Figure. 1. IDT79R3000 Processor

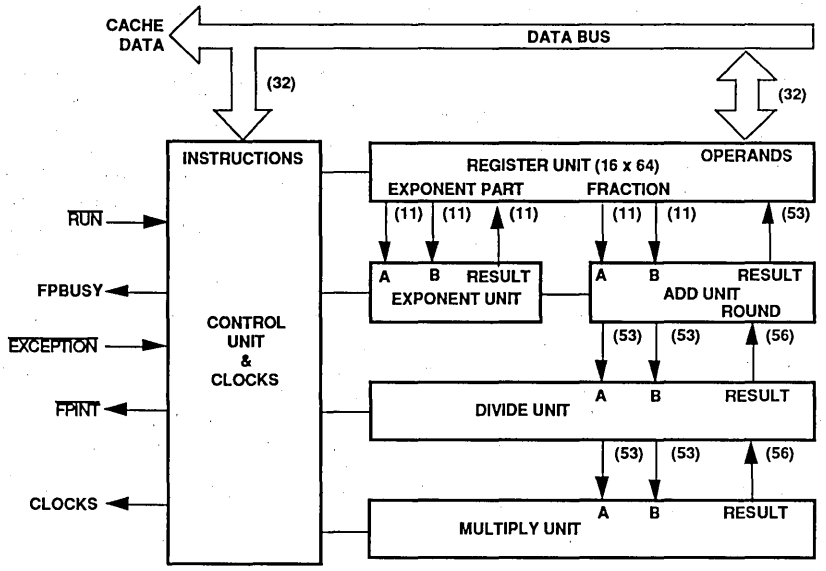


Figure 2. IDT79R3010 Floating-Point Accelerator

THE IDT79R3010 FLOATING-POINT ACCELERATOR

The R3010 Floating-Point Accelerator (FPA) supports full conformance with the IEEE 754 floating-point specification. It acts as a co-processor to the R3000 CPU, providing a seamless integration of fixed and floating-point instructions. All floating-point operations are transparent to the programmer.

The R3010 FPA features:

- Full 64-bit operation
- Single-cycle load/store instructions
- Seamless interface to the R3000 or R3001 CPU
- Three operation units (add/subtract, multiply and divide) can operate in parallel
- Six-level pipeline

THE IDT79R3001 RISController™

The R3001 RISController optimizes the high-performance MIPS architecture for embedded control systems. Capable of 28 MIPS performance at 33MHz, the R3001 incorporates new features for real-time control. The controller extends the performance range of the current R3000 processor, saves valuable real estate for space-critical designs and lowers system memory costs.

The MIPS performance range is extended by the increase in the R3000 in the synchronous memory space from 512KBytes, maximum,, to a full 32MBytes. This allows the system to perform with a guaranteed "cache" hit rate of 100%. The on-chip memory controller allows the designer to use standard SRAMs, DRAMs, or even VRAMs, representing a significant cost savings over other solutions. The processor supports predictable interrupt response times for real-time control applications, and system chip count is lowered by substantially reducing the number of devices needed to implement local memory.

THE R3051 FAMILY OF RISControllers™

The IDT79R3051 Family is a derivative of the R3000, featuring a high level of integration and targeted to high-performance but cost-sensitive embedded processing applications. The R3051 family is designed to bring the high-performance inherent in the MIPS RISC architecture into low-cost, simplified, power sensitive applications.

Functional units were integrated onto the CPU core in order to reduce the total system cost rather than to increase the inherent performance of the integer engine. Thus, the R3051 family is able to offer 35 MIPS of integer performance at 40MHz without requiring external SRAM or caches.

THE R3500A RISCore™ CPU/FPA

The R3500A is a a single chip that integrates the R3000A CPU and the R3010A FPA execution units using the R3000A packaging and pinout. This high integration device is completely binary software, compatible with the R3000, R2000 CPUs and R3010 FPA to facilitate the migration path to higher performance and lower chip count systems that utilize both the CPU and the floating point units.

IDT has also made several enhancements to the R3000 architecture such as faster multiply and divide instructions and

added a programmable tag bus width allowing reduced cache cost. The power consumption is lower by 33% when compared to the standard R3000 and R3010.

THE IDT79R4000 CPU

The R4000 is the third generation of MIPS RISC technology and establishes a new performance standard for RISC processors for the 1990s. The R4000 extends the performance range served by the MIPS architecture and, thereby, provides a migration path to applications served by the R3000, R3001 and the R3051.

This third generation processor maintains full binary compatibility with applications executing on the R2000/R3000 and IDT's RISController family, while achieving substantially higher performance. The key to this performance is both the architecture/implementation of the processor and the high level of integration achieved in a single chip. The R4000 contains the RISC integer unit, floating-point unit, MMU, 8K of I- and D-cache, along with multiprocessing support such as direct control of optional secondary caches. To achieve performance levels capable of over 50 VAX MIPS sustained performance, the R4000 utilizes technology such as super-pipelining to exploit 2 level instruction parallelism with no issue restrictions. The R4000 presents a balanced architectural approach to achieve a wide range of price performance goals.

TABLE OF CONTENTS

Integrated Device Technology, Inc.

RISC CPU PROCESSOR

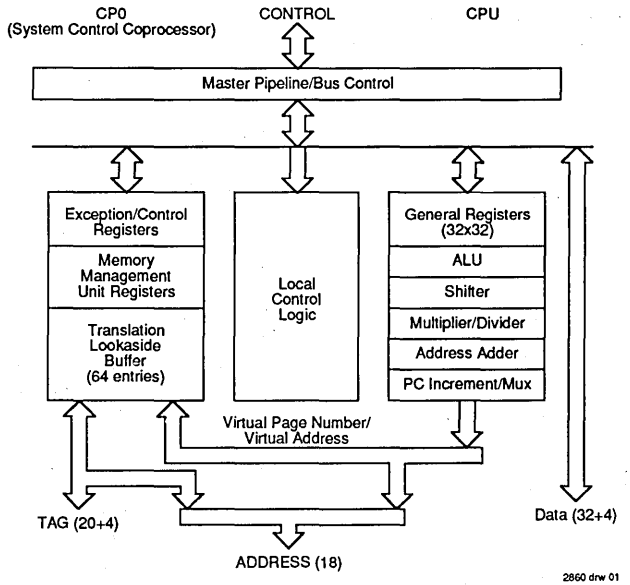
IDT79R3000A
IDT79R3000AE

FEATURES:

- Enhanced instruction set compatible version of the IDT79R2000, IDT79R3000 RISC CPUs.
- Upwardly pin-compatible with IDT79R3000 RISC CPU.
- IDT79R3000A "E" version relaxes system memory timing requirements in a high-speed systems.
- Full 32-bit Operation—Thirty-two 32-bit registers and all instructions and addresses are 32-bit.
- Efficient Pipelining—The CPU's 5-stage pipeline design assists in obtaining an execution rate approaching one instruction per cycle. Pipeline stalls and exceptions are handled precisely and efficiently.
- On-Chip Cache Control—The IDT79R3000A provides a high bandwidth memory interface that handles separate external Instruction and Data Caches ranging in size from 4 to 256 Kbytes each. Both caches are accessed during a single CPU cycle. All cache control is on-chip.
- On-Chip Memory Management Unit—A fully-associative, 64 entry Translation Lookaside Buffer (TLB) provides fast address translation for virtual-to-physical memory mapping of the 4 Gigabyte virtual address space.
- Dynamically able to switch between Big- and Little- Endian byte ordering conventions.
- Coprocessor Interface—The IDT79R3000A generates all addresses and handles memory interface control for up to three additional tightly coupled external processors.
- Optimizing Compilers are available for C, Fortran, Pascal, COBOL, Ada, and PL/1.
- UNIX™ System V.3 and BSD 4.3 operating systems supported.
- High-speed CEMOS™ technology.
- 16.7 through 40MHz clock rates yield up to 32 VUPS sustained throughput.
- Supports independent multiword block refill of both the instruction and data caches with variable block sizes.
- Supports concurrent refill and execution of instructions.
- Partial word stores executed as read-modify-write operations.
- 6 external interrupt inputs, 2 software interrupts, with single cycle latency to exception handler routine.
- Flexible multiprocessing support on chip with no impact on uniprocessor designs.
- Military product compliant to MIL-STD-883, Class B.

5

IDT79R3000A PROCESSOR



CEMOS is a trademark of Integrated Device Technology, Inc.

DESCRIPTION:

The IDT 79R3000A RISC Microprocessor consists of two tightly-coupled processors integrated on a single chip. The first processor is a full 32-bit CPU based on RISC (Reduced Instruction Set Computer) principles to achieve a new standard of microprocessor performance. The second processor is a system control coprocessor, called CP0, containing a fully-associative 64 entry TLB (Translation Lookaside Buffer), MMU (Memory Management Unit) and control registers, supporting a 4 Gigabyte virtual memory subsystem, and a Harvard Architecture Cache Controller achieving a bandwidth of 320 Mbytes/second using industry standard static RAMs.

This data sheet provides an overview of the features and architecture of the 79R3000A CPU, Revision 3.0. A more detailed description of the operation of the device is incorporated in the "R3000A Family Hardware User Manual", and a more detailed architectural overview is provided in the "mips RISC Architecture" book, both available from IDT. Documentation providing details of the software and development environments supporting this processor are also available from IDT.

IDT79R3000A CPU Registers

The IDT79R3000A CPU provides 32 general purpose 32-bit registers, a 32-bit Program Counter, and two 32-bit registers that hold the results of integer multiply and divide operations. Only two of the 32 general registers have a special purpose: register r0 is hardwired to the value "0", which is a useful constant, and register r31 is used as the link register in jump-and-link instructions (return address for subroutine calls).

The CPU registers are shown in Figure 2. Note that there is no Program Status Word (PSW) register shown in this figure: the functions traditionally provided by a PSW register are instead provided in the Status and Cause registers incorporated within the System Control Coprocessor (CP0).

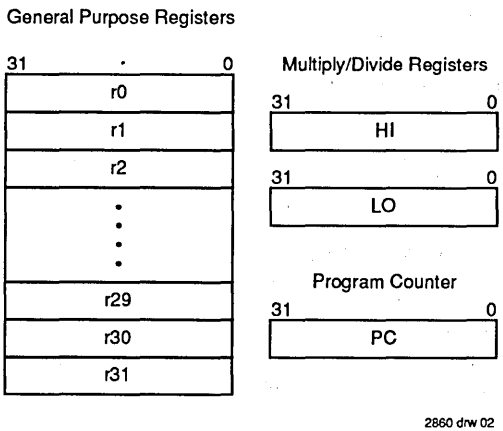


Figure 2. IDT79R3000A CPU Registers

Instruction Set Overview

All IDT79R3000A instructions are 32 bits long, and there are only three instruction formats. This approach simplifies instruction decoding, thus minimizing instruction execution time. The 79R3000A processor initiates a new instruction on every run cycle, and is able to complete an instruction on almost every clock cycle. The only exceptions are the Load instructions and Branch instructions, which each have a single cycle of latency associated with their execution. Note, however, that in the majority of cases the compilers are able to fill these latency cycles with useful instructions which do not require the result of the previous instruction. This effectively eliminates these latency effects.

The actual instruction set of the CPU was determined after extensive simulations to determine which instructions should be implemented in hardware, and which operations are best synthesized in software from other basic instructions. This methodology resulted in the R3000A having the highest performance of any available microprocessor.

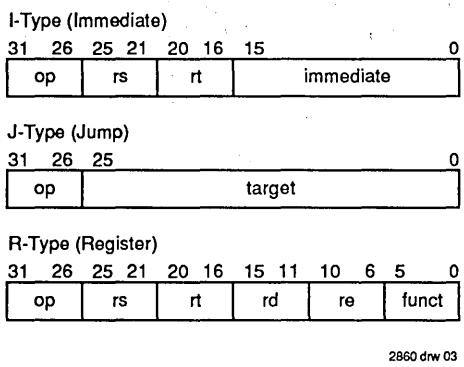


Figure 3. IDT79R3000A Instruction Formats

The IDT79R3000A instruction set can be divided into the following groups:

- **Load/Store** instructions move data between memory and general registers. They are all I-type instructions, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.
  The Load instruction has a single cycle of latency, which means that the data being loaded is not available to the instruction immediately after the load instruction. The compiler will fill this delay slot with either an instruction which is not dependent on the loaded data, or with a NOP instruction. There is no latency associated with the store instruction.
  Loads and Stores can be performed on byte, haff-word, word, or unaligned word data (32 bit data not aligned on a modulo-4 address). The CPU cache is constructed as a write-through cache.
- **Computational** instructions perform arithmetic, logical and shift operations on values in registers. They occur in both R-type (both operands and the result are registers)

and I-type (one operand is a 16-bit immediate) formats. Note that computational instructions are three operand instructions; that is, the result of the operation can be stored into a different register than either of the two operands. This means that operands need not be overwritten by arithmetic operations. This results in a more efficient use of the large register set.

- **Jump and Branch** instructions change the control flow of a program. Jumps are always to a paged absolute address formed by combining a 26-bit target with four bits of the Program counter (J-type format, for subroutine calls), or 32-bit register byte addresses (R-type, for returns and dispatches). Branches have 16-bit offsets relative to the program counter (I-type). Jump and Link instructions save a return address in Register 31. The 79R3000A instruction set features a number of branch conditions. Included is the ability to compare a register to zero and branch, and also the ability to branch based on a comparison between two registers. Thus, net performance is increased since software does not have to perform arithmetic instructions prior to the branch to set up the branch conditions.
- **Coprocessor** instructions perform operations in the coprocessors. Coprocessor Loads and Stores are I-type. Coprocessor computational instructions have coprocessor-dependent formats (see coprocessor manuals).
- **Coprocessor 0** instructions perform operations on the System Control Coprocessor (CP0) registers to manipulate the memory management and exception handling facilities of the processor.
- **Special** instructions perform a variety of tasks, including movement of data between special and general registers, system calls, and breakpoint. They are always R-type.

Table 1 lists the instruction set of the IDT79R3000A processor.

5

OP	Description	OP	Description
	Load/Store Instructions		**Multiply/Divide Instructions**
LB	Load Byte	MULT	Multiply
LBU	Load Byte Unsigned	MULTU	Multiply Unsigned
LH	Load Halfword	DIV	Divide
LHU	Load Halfword Unsigned	DIVU	Divide Unsigned
LW	Load Word	MFHI	Move From HI
LWL	Load Word Left	MTHI	Move To HI
LWR	Load Word Right	MFLO	Move From LO
SB	Store Byte	MTLO	Move To LO
SH	Store Halfword		
SW	Store Word		**Jump and Branch Instructions**
SWL	Store Word Left	J	Jump
SWR	Store Word Right	JAL	Jump and Link
		JR	Jump to Register
	Arithmetic Instructions	JALR	Jump and Link Register
	(ALU Immediate)	BEQ	Branch on Equal
ADDI	Add Immediate	BNE	Branch on Not Equal
ADDIU	Add Immediate Unsigned	BLEZ	Branch on Less than or Equal to Zero
SLTI	Set on Less Than Immediate	BGTZ	Branch on Greater Than Zero
SLTIU	Set on Less Than Immediate Unsigned	BLTZ	Branch on Less Than Zero
		BGEZ	Branch on Greater than or Equal to Zero
ANDI	AND Immediate		
ORI	OR Immediate	BLTZAL	Branch on Less Than Zero and Link
XORI	Exclusive OR Immediate	BGEZAL	Branch on Greater than or Equal to Zero and Link
LUI	Load Upper Immediate		
			Special Instructions
	Arithmetic Instructions	SYSCALL	System Call
	(3-operand, register-type)	BREAK	Break
ADD	Add		
ADDU	Add Unsigned		**Coprocessor Instructions**
SUB	Subtract	LWCz	Load Word from Coprocessor
SUBU	Subtract Unsigned	SWCz	Store Word to Coprocessor
SLT	Set on Less Than	MTCz	Move To Coprocessor
		MFCz	Move From Coprocessor
SLTU	Set on Less Than Unsigned	CTCz	Move Control to Coprocessor
AND	AND	CFCz	Move Control From Coprocessor
OR	OR	COPz	Coprocessor Operation
XOR	Exclusive OR	BCzT	Branch on Coprocessor z True
NOR	NOR	BCzF	Branch on Coprocessor z False
	Shift Instructions		**System Control Coprocessor**
SLL	Shift Left Logical		**(CP0) Instructions**
SRL	Shift Right Logical	MTC0	Move To CP0
SRA	Shift Right Arithmetic	MFC0	Move From CP0
SLLV	Shift Left Logical Variable	TLBR	Read indexed TLB entry
SRLV	Shift Right Logical Variable	TLBWI	Write Indexed TLB entry
SRAV	Shift Right Arithmetic Variable	TLBWR	Write Random TLB entry
		TLBP	Probe TLB for matching entry
		RFE	Restore From Exception

2860 tbl 01

Table 1. IDT79R3000A Instruction Summary

IDT79R3000A System Control Coprocessor (CP0)

The IDT79R3000A can operate with up to four tightly-coupled coprocessors (designated CP0 through CP3). The System Control Coprocessor (or CP0), is incorporated on the IDT79R3000 chip and supports the virtual memory system and exception handling functions of the IDT79R3000A. The virtual memory system is implemented using a Translation Lookaside Buffer and a group of programmable registers as shown in Figure 4.

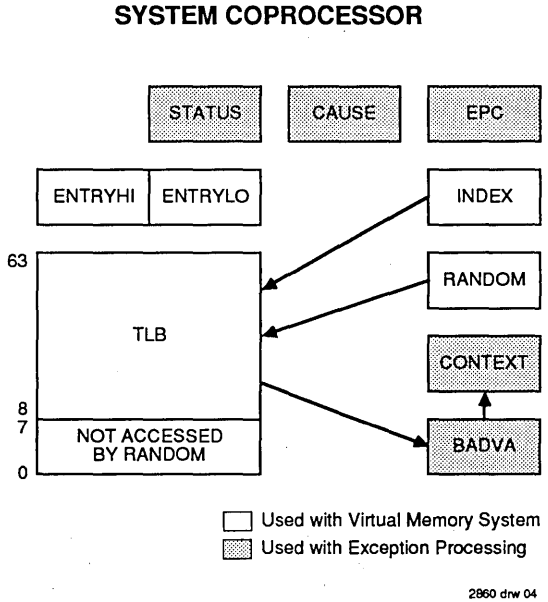
System Control Coprocessor (CP0) Registers

The CP0 registers shown in Figure 4 are used to control the memory management and exception handling capabilities of the IDT79R3000A. Table 2 provides a brief description of each register.

Register	Description
EntryHi	High half of a TLB entry
EntryLo	Low half of a TLB entry
Index	Programmable pointer into TLB array
Random	Pseudo-random pointer into TLB array
Status	Mode, interrupt enables, and diagnostic status info
Cause	Indicates nature of last exception
EPC	Exception Program Counter
Context	Pointer into kernel's virtual Page Table Entry array
BadVA	Most recent bad virtual address
PRId	Processor revision identification (Read only)

2860 tbl 02

Table 2. System Control Coprocessor (CP0) Registers



2860 drw 04

Figure 4. The System Coprocessor Registers

5

Memory Management System

The IDT79R3000A has an addressing range of 4 Gbytes. However, since most IDT79R3000A systems implement a physical memory smaller than 4Gbytes, the IDT79R3000A provides for the logical expansion of memory space by translating addresses composed in a large virtual address space into available physical memory address. The 4 GByte address space is divided into 2 GBytes which can be accessed by both the users and the kernel, and 2 GBytes for the kernel only.

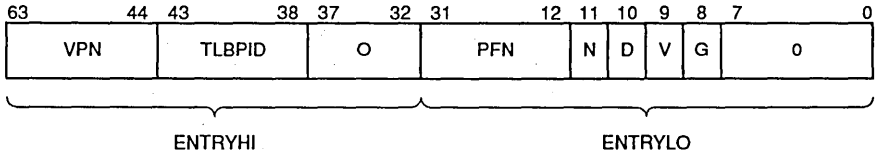
The TLB (Translation Lookaside Buffer)

Virtual memory mapping is assisted by the Translation Lookaside Buffer (TLB). The on-chip TLB provides very fast virtual memory access and is well-matched to the requirements of multi-tasking operating systems. The fully-associative TLB contains 64 entries, each of which maps a 4-Kbyte page, with controls for read/write access, cacheability, and process identification. The TLB allows each user to access up to 2 Gbytes of virtual address space.

Figure 5 illustrates the format of each TLB entry. The Translation operation involves matching the current Process ID (PID) and upper 20 bits of the address against PID and VPN (Virtual Page Number) fields in the TLB. When both match (or the TLB entry is Global), the VPN is replaced with the PFN (Physical Frame Number) to form the physical address.

TLB misses are handled in software, with the entry to be replaced determined by as imple RANDOM function. The routine to process a TLB miss in the UNIX environment requires only 10-12 cycles, which compares favorably with many CPUs which perform the operation in hardware.

TLB ENTRY FORMAT



VPN – Virtual Page Number
TLBPID – Process ID
PFN – Physical Frame Number
N – Non-cacheable flag
D – Dirty flag (Write protect)
V – Valid entry flag
G – Global flag (ignore PID)
0 – Reserved

2860 drw 05

Figure 5. TLB Entry Format

IDT79R3000A Operating Modes

The IDT79R3000A has two operating modes: User mode and Kernel/mode. The IDT79R3000A normally operates in the User mode until an exception is detected forcing it into the Kernel mode. It remains in the Kernel mode until a Restore From Exception (RFE) instruction is executed. The manner in which memory addresses are translated or mapped depends on the operating mode of the IDT79R3000A. Figure 6 shows the MMU translation performed for each of the operating modes.

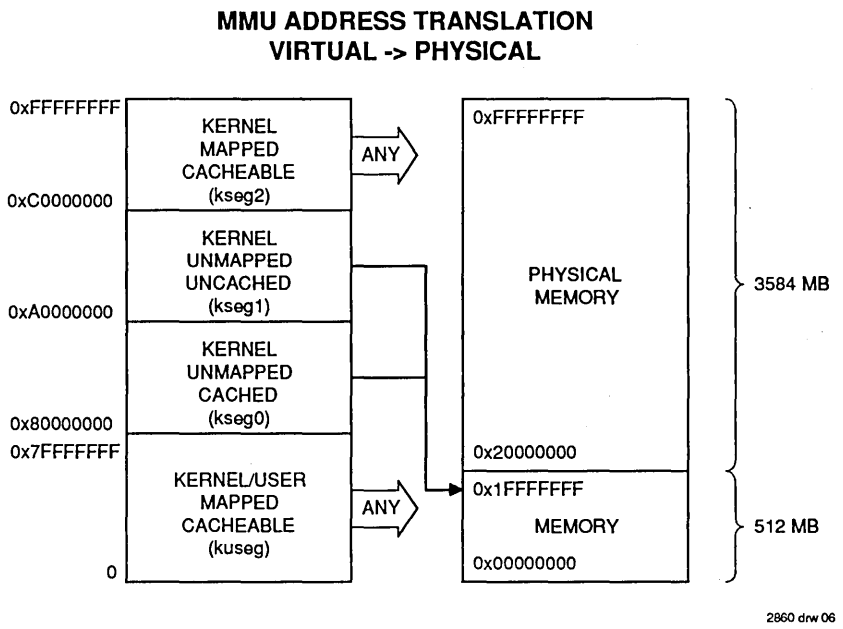


Figure 6. IDT79R3000A Virtual Address Mapping

User Mode—in this mode, a single, uniform virtual address space (kuseg) of 2 Gbyte is available. Each virtual address is extended with a 6-bit process identifier field to form unique virtual addresses. All references to this segment are mapped through the TLB. Use of the cache for up to 64 processes is determined by bit settings for each page within the TLB entries.

Kernel Mode—four separate segments are defined in this mode:

- *kuseg*—when in the kernel mode, references to this segment are treated just like user mode references, thus streamlining kernel access to user data.
- *kseg0*—references to this 512 Mbyte segment use cache memory but are not mapped through the TLB. Instead, they always map to the first 0.5 GBytes of physical address space.
- *kseg1*—references to this 512 Mbyte segment are not mapped through the TLB and do not use the cache. Instead, they are hard-mapped into the same 0.5 GByte segment of physical address space as kseg0.
- *kseg2*—references to this 1 Gbyte segment are always mapped through the TLB and use of the cache is determined by bit settings within the TLB entries.

IDT79R3000 Pipeline Architecture

The execution of a single IDT79R3000A instruction consists of five primary steps:

1) **IF** — Fetch the instruction (I-Cache).
2) **RD** — Read any required operands from CPU registers while decoding the instruction.
3) **ALU** — Perform the required operation on instruction operands.
4) **MEM** — Access memory (D-Cache).
5) **WB** — Write back results to register file.

Each of these steps requires approximately one CPU cycle as shown in Figure 7 (parts of some operations overlap into another cycle while other operations require only 1/2 cycle).

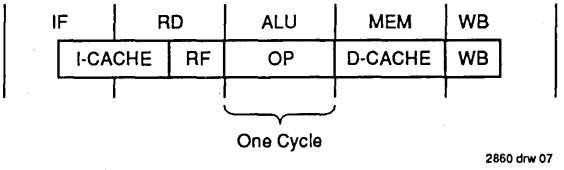


Figure 7. IDT79R3000A Instruction Pipeline

INSTRUCTION EXECUTION

The IDT79R3000A uses a 5-stage pipeline to achieve an instruction execution rate approaching one instruction per CPU cycle. Thus, execution of five instructions at a time are overlapped as shown in Figure 8.

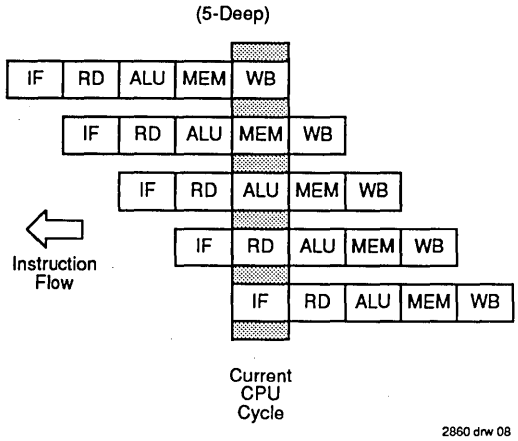


Figure 8. IDT79R3000A Execution Sequence

IDT79R3000A INSTRUCTION PIPELINE

This pipeline operates efficiently because different CPU resources (address and data bus accesses, ALU operations, register accesses, and so on) are utilized on a non-interfering basis.

Memory System Hierarchy

The high performance capabilities of the IDT79R3000A processor demand system configurations incorporating techniques frequently employed in large, mainframe computers but seldom encountered in systems based on more traditional microprocessors.

A primary goal of systems employing RISC techniques is to minimize the average number of cycles each instruction requires for execution. In order to achieve this goal, RISC processors incorporate a number of RISC techniques, including a compact and uniform instruction set, a deep instruction pipeline (as described above), and utilization of optimizing compilers. Many of the advantages obtained from these techniques can, however, be negated by an inefficient memory system.

Figure 9 illustrates memory in a simple microprocessor system. In this system, the CPU outputs addresses to memory and reads instructions and data from memory or writes data to memory. The address space is completely undifferentiated: instructions, data, and I/O devices are all treated the same. In such a system, a primary limiting performance factor is memory bandwidth.

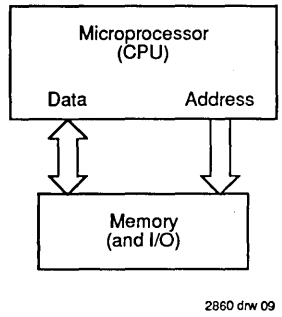


Figure 9. A Simple Microprocessor Memory System

Figure 10 illustrates a memory system that supports the significantly greater memory bandwidth required to take full advantage of the IDT79R3000A's performance capabilities. The key features of this system are:

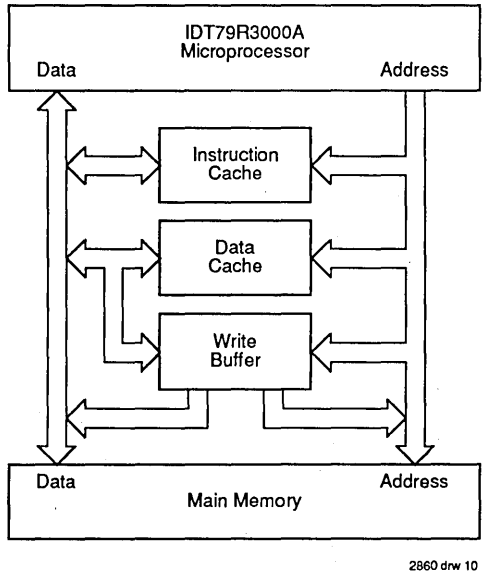


Figure 10. An IDT79R3000A System with a High-Performance Memory System

- **External Cache Memory**—Local, high-speed memory (called cache memory) is used to hold instructions and data that is repetitively accessed by the CPU (for example, within a program loop) and thus reduces the number of references that must be made to the slower-speed main memory. Some microprocessors provide a limited amount of cache memory on the CPU chip itself. The external caches supported by the IDT79R3000A can be much larger; while a small cache can improve performance of some programs, significant improvements for a wide range of programs require large caches.
- **Separate Caches for data and Instructions**—Even with high-speed caches, memory speed can still be a limiting factor because of the fast cycle time of a high-performance microprocessor. The IDT79R3000A supports separate caches for instructions and data and alternates accesses of the two caches during each CPU cycle. Thus, the processor can obtain data and instructions at the cycle rate of the CPU using caches constructed with commercially available IDT static RAM devices.
  In order to maximize bandwidth in the cache while minimizing the requirement for SRAM access speed, the R3000A divides a single-processor clock cycle into two phases. During one phase, the address for the data cache access is presented while data previously addressed in the instruction cache is read; during the next phase, the data operation is completed while the instruction cache is being addressed. Thus, both caches are read in a single processor cycle using only one set of address and data pins.

- **Write Buffer**—in order to ensure data consistency, all data that is written to the data cache must also be written out to main memory. The cache write model used by the IDT79R3000A is that of a write-through cache; that is, all data written by the CPU is immediately written into the main memory. To relieve the CPU of this responsibility (and the inherent performance burden) the IDT79R3000A supports an interface to a write buffer. The IDT79R3020 Write Buffer captures data (and associated addresses) output by the CPU and ensures that the data is passed on to main memory.

IDT79R3000A Processor Subsystem Interfaces

Figure 11 illustrates the three subsystem interfaces provided by the IDT79R3000A processor:

- **Cache control** interface (on-chip) for separate data and instruction caches permits implementation of off-chip caches using standard IDT SRAM devices. The 79R3000A directly controls the cache memory with a minimum of external components. Both the instruction and data cache can vary from 0 to 256K Bytes (64K entries). The 79R3000A also includes the TAG control logic which determines whether or not the entry read from the cache is the desired data. The 79R3000A cache controller implements a direct mapped cache for high net performance (bandwidth). It has the ability to refill multiple words when a cache miss occurs, thus reducing the effective miss rate to less than 2% for large caches. When a cache miss occurs, the 79R3000A can support refilling the cache in 1, 4, 8, 16, or 32 word blocks to minimize the effective penalty of having to access main memory. The 79R3000A also incorporates the ability to perform instruction streaming; while the cache is refilling, the processor can resume execution once the missed word is obtained from main memory. In this way, the processor can continue to execute concurrently with the cache block refill.
- **Memory controller** interface for system (main) memory. This interface also includes the logic and signals to allow operation with a write buffer to further improve memory bandwidth. In addition to the standard full word access, the memory controller supports the ability to write bytes and half-words by using partial word operations. The memory controller also supports the ability to retry memory accesses if, for example, the data returned from memory is invalid and a bus error needs to be signalled.
- **Coprocessor Interface**—The IDT79R3000A features a tightly coupled co-processor interface in which all coprocessors maintain synchronization with the main processor; reside on the same data bus as the main processor; and participate in bus transactions in an identical manner to the main processor. The IDT79R3000A generates all required cache and memory control signals, including cache and memory addresses for attached coprocessors. As a result, only the data bus and a few control signals need to be connected to a coprocessor.
  The interface supports three types of coprocessor instructions: loads/stores, coprocessor operations, and processor-coprocessor transfers. Note that coprocessor loads and stores occur directly between the coprocessor and memory, without requiring the data to go through the CPU. Synchronization between the CPU and external coprocessors is achieved using a Phased-Lock Loop interface to the coprocessor. The coprocessor physical interface also includes coprocessor condition signals (CpCond(n)), which are used in coprocessor branch instructions, and a coprocessor busy signal (CpBusy) which is used to stall the CPU if the coprocessor needs to hold off subsequent operations.
  Finally, a precise exception interface is defined between the CPU and coprocessors using the external interrupt inputs of the CPU. This allows a coprocessor exception, even if it was the result of a multi-cycle operation, to be traced to the precise coprocessor operation which caused it. This is an important feature for languages which can define specific error handlers for each task.
  The interface supports up to four separate coprocessors. Coprocessor 0 is defined to be the system control coprocessor, and resides on the same chip as the CPU unit. Coprocessor 1 is the Floating Point Accelerator, IDT 79R301 0A. Coprocessors 2 and 3 are available to support an interface to application specific functions.

5

MULTIPROCESSING SUPPORT

The IDT79R3000A supports multiprocessing applications in a simple but effective way. Multiprocessing applications require cache coherency across the multiple processors. The IDT79R3000A offers two signals to support cache coherency: the first, MPStall, stalls the processor within two cycles of being received and keeps it from accessing the cache. This allows an external agent to snoop into the processor data cache. The second signal, MPInvalidate, causes the processor to write data on the data cache bus which indicates the externally addressed cache entry is invalid. Thus, a subsequent access to that location would result in a cache miss, and the data would be obtained from main memory.

The two MP signals would be generated by a external logic which utilizes a secondary cache to perform bus snooping functions. The 79R3000A does not impose an architecture for this secondary cache, but rather is flexible enough to support a variety of application specific architecture stand still maintain cache coherency. Further, there is no impact on designs which do not require this feature. The 79R3000A has further improved on the microprocessor support found in the 79R3000, by allowing the use of cache RAMs with internal address latches in multiprocessor systems.

ADVANCED FEATURES

The IDT79R3000A offers a number of additional features such as the ability to swap the instruction and data caches, facilitating diagnostics and cache flushing. Another feature isolates the caches, which forces cache hits to occur regardless of the contents of the tag fields. The IDT79R3000A allows the processor to execute user tasks of the opposite byte ordering (endianness) of the operating system, and further allows parity checking to be disabled. More details on these features can be found in the IDT 79R3000A Family Hardware User's Manual.

Further features of the IDT79R3000A are configured during the last four cycles prior to the negation of the RESET input. These functions include the ability to select cache sizes and cache refill block sizes; the ability to utilize the multiprocessor interface; whether or not instruction streaming is enabled; whether byte ordering follows "Big-Endian" or "Little-Endian" protocols, etc. Table 3 shows the configuration options selected at Reset. These are further discussed in the "Hardware User's Manual".

BACKWARD COMPATIBILITY WITH 79R2000

The IDT79R3000A can be used in sockets designed for the 79R3000. The pin-out of the 79R3000A has been selected to ensure this compatibility, with new functions mapped onto previously unused pins. The instruction set is compatible with that of the 79R2000 at the binary level. As a result, code written for the older processor can be executed. New features can be selectively disabled.

In most 79R3000 applications, the 79R3000A can be placed in the socket with no modification to initialization settings. Further application assistance on this topic is available from IDT.

PACKAGE THERMAL SPECIFICATIONS

The IDT79R3000A utilizes special packaging techniques to improve both the thermal and electrical characteristics of the microprocessor.

In order to improve the electrical characteristics of the device, the package is constructed using multiple signal planes, including individual power planes and ground planes to reduce noise associated with high-frequency TTL parts. In addition, the 175-pin PGA package utilizes extra power and ground pins to reduce the inductance from the internal power planes to the power planes of the PC Board.

In order to improve the electrical characteristics of the microprocessor, the device is housed using cavity down packaging. In addition, these packages incorporate a copper-tungsten thermal slug designed to efficiently transfer heat from the die to the case of the package, and thus effectively lower the thermal resistance of the package. The use of an additional external heat sink affixed to the package thermal slug further decreases the effective thermal resistance of the package.

The case temperature may be measured in any environment to determine whether the device is within the specified operating range. The case temperature should be measured at the center of the top surface opposite the package cavity (the package cavity is the side where the package lid is mounted).

The equivalent allowable ambient temperature, T_A, can be calculated using the thermal resistance from case to ambient (Øca) for the given package. The following equation relates ambient and case temperature:

$$T_A = T_c - P * Øca$$

where P is the maximum power consumption, calculated by using the maximum Icc from the DC Electrical Characteristics section.

Typical values for Øca at various airflows are shown in table 4 for the various CPU packages.

	Airflow - (ft/min)					
	0	200	400	600	800	1000
Øca (175-PGA, 144-PGA)	21	7	3	2	1	0.5
Øca (172 Quad Flatpack)	23	9	4	3	2.5	1.5

2860 tbl 03

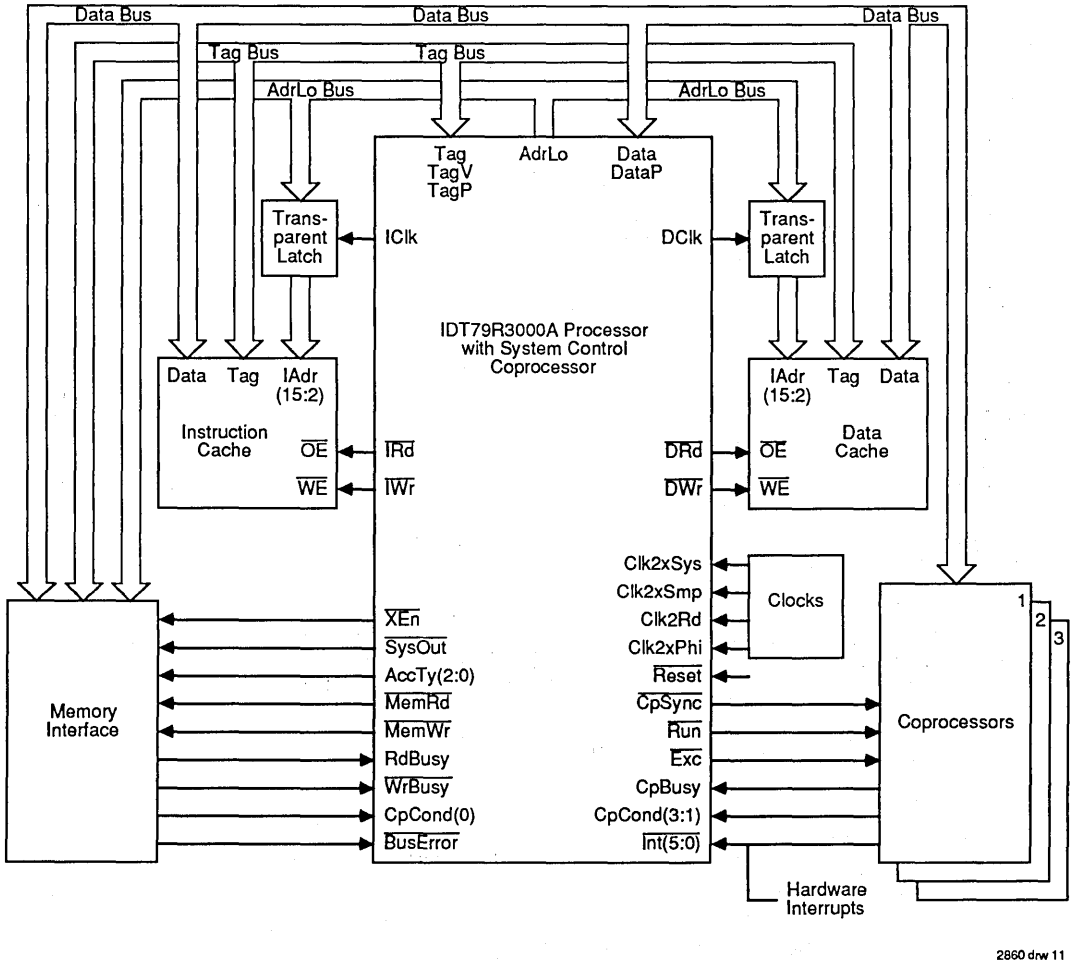
Table 4. R3000A Package Characteristics

Input	W Cycle	X Cycle	Y Cycle	Z Cycle
Int0	DBlkSize0	DBlkSize1	Extend Cache	Big Endian
Int1	IBlkSize0	IBlkSize1	MPAdrDisable	TriState
Int2	DispPar/RevEnd	IStream	IgnoreParity	NoCache
Int3	Reserved(1)	StorePartial	MultiProcessor	BusDriveOn
Int4	PhaseDelayOn(2)	PhaseDelayOn(2)	PhaseDelayOn(2)	PhaseDelayOn(2)
Int5	R3000 Mode(2)	R3000 Mode(2)	R3000 Mode(2)	R3000 Mode(2)

2860 tbl 04

NOTES:
1. Reserved entries must be driven high.
2. These values must be driven stable throughout the entire RESET period.

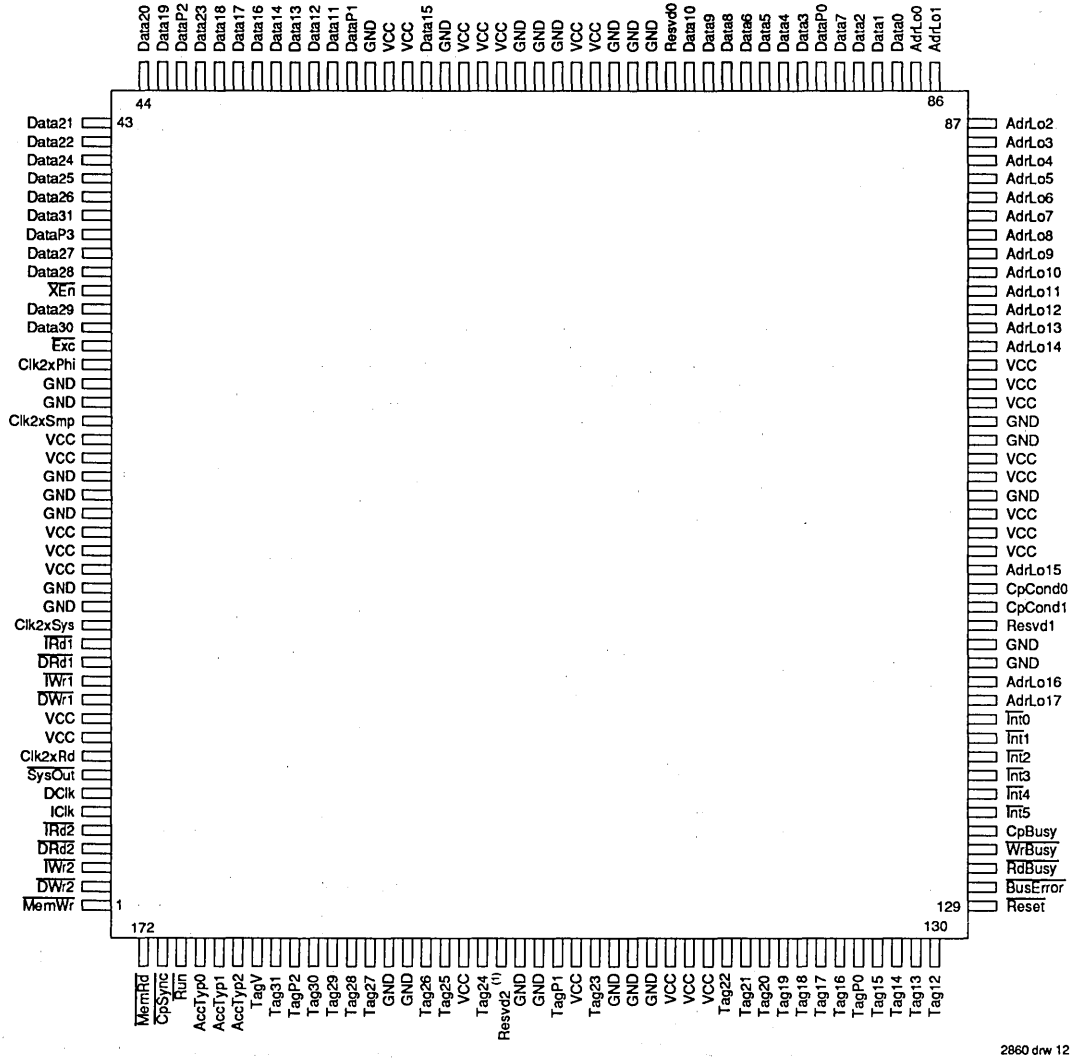
Table 3. R3000A Mode Selectable Features



2860 drw 11

Figure 11. IDT79R3000A Subsystem Interfaces Example; 64 KB Caches

PIN CONFIGURATION



2860 drw 12

172-Pin Flatpack (Top View)

NOTES:
1. Reserved pins must be connected.
2. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN CONFIGURATION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	(No Pin)	AdrLo 6	AdrLo 10	AdrLo 11	VCC	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	$\overline{\text{Int(2)}}$	$\overline{\text{Int(5)}}$	Wr Busy	$\overline{\text{Reset}}$	VCC
B	AdrLo 3	$\overline{\text{DRd2}}$	AdrLo 7	AdrLo 9	AdrLo 12	$\overline{\text{IRd2}}$	AdrLo 13	CpCond 1	$\overline{\text{Int(1)}}$	$\overline{\text{Int(3)}}$	Cp Busy	$\overline{\text{Bus Error}}$	$\overline{\text{DWr2}}$	Tag12	Tag15
C	AdrLo 0	AdrLo 4	VCC	AdrLo 5	AdrLo 8	GND	GND	VCC	$\overline{\text{Int(0)}}$	$\overline{\text{Int(4)}}$	Rd Busy	GND	Tag13	TagP0	Tag18
D	Data 1	AdrLo 2	GND	GND	VCC	GND	VCC	GND	VCC	GND	VCC	GND	Tag14	Tag17	Tag19
E	DataP 0	Data 0	AdrLo 1	VCC								VCC	Tag16	Tag20	VCC
F	VCC	Data 7	Data 2	GND								GND	GND	Tag21	Tag23
G	Data 4	Data 3	GND	VCC								VCC	GND	Tag22	TagP1
H	Data 6	Data 5	Data 8	GND								GND	VCC	Tag25	Tag24
J	Data 10	DataP 1	Data 9	VCC								VCC	Tag28	Tag29	Tag26
K	Data 15	Data 11	GND	GND								GND	GND	TagP2	Tag27
L	VCC	Data 12	Data 17	VCC								VCC	Acc Typ2	Tag31	Tag30
M	Data 13	Data 16	DataP 2	GND	VCC	GND	VCC	GND	VCC	GND	VCC	GND	GND	Acc Typ1	VCC
N	Data 14	Data 18	Data 19	GND	Data 24	DataP 3	VCC	VCC	GND	GND	$\overline{\text{DRd1}}$	$\overline{\text{Mem Wr}}$	$\overline{\text{Mem Rd}}$	$\overline{\text{Run}}$	TagV
P	Data 23	Data 20	$\overline{\text{IWr2}}$	Data 22	Data 26	Data 27	$\overline{\text{XEn}}$	Data 30	Clk2x Sys	Clk2x Rd	DClk	$\overline{\text{IRd1}}$	$\overline{\text{IWr1}}$	$\overline{\text{Cp Sync}}$	Acc Typ0
Q	VCC	Data 21	Data 25	Data 31	Data 28	GND	Data 29	$\overline{\text{Excep-tion}}$	Clk2x Phi	Clk2x Smp	$\overline{\text{SysOut}}$	VCC	IClk	$\overline{\text{DWr1}}$	VCC

2860 drw 13

175-Pin PGA (Top View)

NOTE:
1. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN CONFIGURATION

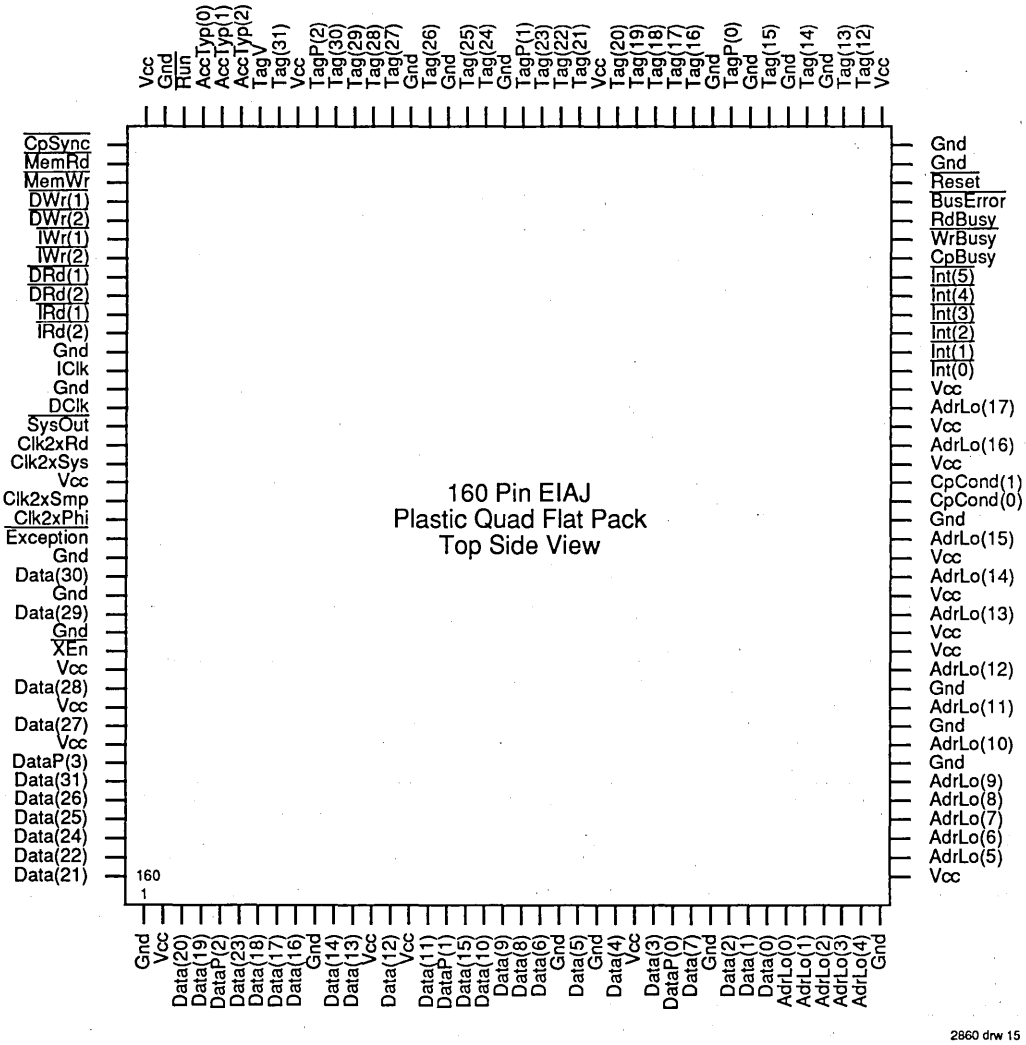
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	VCC	AdrLo 6	AdrLo 10	AdrLo 11	VCC	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	$\overline{\text{Int(2)}}$	$\overline{\text{Int(5)}}$	Wr Busy	$\overline{\text{Reset}}$	VCC
B	AdrLo 3	$\overline{\text{DRd2}}$	AdrLo 7	AdrLo 9	AdrLo 12	$\overline{\text{IRd2}}$	AdrLo 13	CpCond 1	$\overline{\text{Int(1)}}$	$\overline{\text{Int(3)}}$	Cp Busy	$\overline{\text{Bus Error}}$	$\overline{\text{DWr2}}$	Tag12	Tag15
C	AdrLo 0	AdrLo 4	VCC	AdrLo 5	AdrLo 8	GND	GND	VCC	$\overline{\text{Int(0)}}$	$\overline{\text{Int(4)}}$	Rd Busy	GND	Tag13	TagP0	Tag18
D	Data 1	AdrLo 2	GND	GND									Tag14	Tag17	Tag19
E	DataP 0	Data 0	AdrLo 1										Tag16	Tag20	VCC
F	VCC	Data 7	Data 2										GND	Tag21	Tag23
G	Data 4	Data 3	GND										GND	Tag22	TagP1
H	Data 6	Data 5	Data 8										VCC	Tag25	Tag24
J	Data 10	DataP 1	Data 9										Tag28	Tag29	Tag26
K	Data 15	Data 11	GND										GND	TagP2	Tag27
L	VCC	Data 12	Data 17										Acc Typ2	Tag31	Tag30
M	Data 13	Data 16	DataP 2										GND	Acc Typ1	VCC
N	Data 14	Data 18	Data 19	GND	Data 24	DataP 3	VCC	VCC	GND	GND	$\overline{\text{DRd1}}$	$\overline{\text{Mem Wr}}$	$\overline{\text{Mem Rd}}$	$\overline{\text{Run}}$	TagV
P	Data 23	Data 20	$\overline{\text{IWr2}}$	Data 22	Data 26	Data 27	$\overline{\text{XEn}}$	Data 30	Clk2x Sys	Clk2x Rd	DClk	$\overline{\text{IRd1}}$	$\overline{\text{IWr1}}$	$\overline{\text{Cp Sync}}$	Acc Typ0
Q	VCC	Data 21	Data 25	Data 31	Data 28	GND	Data 29	$\overline{\text{Excep-tion}}$	Clk2x Phi	Clk2x Smp	$\overline{\text{SysOut}}$	VCC	IClk	$\overline{\text{DWr1}}$	VCC

2860 drw 14

144-Pin PGA (Top View)

NOTE:
1. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN CONFIGURATION



2860 drw 15

NOTE:

1. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN DESCRIPTIONS

Pin Name	I/O	Description
Data (0-31)	I/O	A 32-bit bus used for all instruction and data transmission among the processor, caches, memory interface, and coprocessors.
DataP (0-3)	I/O	A 4-bit bus containing even parity over the data bus.
Tag (12-31)	I/O	A 20-bit bus used for transferring cache tags and high addresses between the processor, caches, and memory interface.
TagV	I/O	The tag validity indicator.
Tag P (0-2)	I/O	A 3-bit bus containing even parity over the concatenation of TagV and Tag.
AdrLo (0-17)	O	An 18-bit bus containing byte addresses used for transferring low addresses from the processor to the caches and memory interface. (AdrLo 16: CpCond (2), AdrLo 17: CpCond (3) set by reset initialization).
$\overline{\text{IRd1}}$	O	Read enable for the instruction cache.
$\overline{\text{IWr1}}$	O	Write enable for the instructon cache.
$\overline{\text{IRd2}}$	O	An identical copy of $\overline{\text{IRd1}}$ used to split the load.
$\overline{\text{IWr2}}$	O	An identical copy of $\overline{\text{IWr1}}$ used to split the load.
IClk	O	The instruction cache address latch clock. This clock runs continuously.
$\overline{\text{DRd1}}$	O	The read enable for the data cache.
$\overline{\text{DWr1}}$	O	The write enable for the data cache.
$\overline{\text{DRd2}}$	O	An identical copy of $\overline{\text{DRd1}}$ used to split the load.
$\overline{\text{DWr2}}$	O	An identical copy of $\overline{\text{DWr1}}$ used to split the load.
DClk	O	The data cache address latch clock. This clock runs continuously.
$\overline{\text{XEn}}$	O	The read enable for the Read Buffer.
AccTyp(0-2)	O	A 3-bit bus used to indicate the size of data being transferred on the data bus, whether or not a data transfer is occurring, and the purpose of the transfer.
$\overline{\text{MemWr}}$	O	Signals the occurrence of a main memory write.
$\overline{\text{MemRd}}$	O	Signals the occurrence of a main memory read.
$\overline{\text{BusError}}$	I	Signals the occurrence of a bus error during a main memory read or write.
$\overline{\text{Run}}$	O	Indicates whether the processor is in the run or stall state.
$\overline{\text{Exception}}$	O	Indicates that the instruction about to commit state should be aborted and other exception related information.
$\overline{\text{SysOut}}$	O	A reflection of the internal processor clock used to generate the system clock.
$\overline{\text{CpSync}}$	O	A clock which is identical to $\overline{\text{SysOut}}$ and used by coprocessors for timing synchronization with the CPU.
RdBusy	I	The main memory read stall termination signal. In most system designs RdBusy is normally asserted and is deasserted only to indicate the successful completion of a memory read. RdBusy is sampled by the processor only during memory read stalls.
$\overline{\text{WrBusy}}$	I	The main memory write stall initiation/termination signal.
CpBusy	I	The coprocessor busy stall initiation/termination signal.
CpCond (0-1)	I	A 2-bit bus used to transfer conditional branch status from the coprocessors to the main processor.
CpCond (2-3)	I	Conditional branch status from coprocessors to the processor. Function is provided on AdrLo 16/17 pins and is selected at reset time.
MPStall	I	Multiprocessing Stall. Signals to the processor that it should stall accesses to the caches in a multiprocessing environment. This is physically the same pin as CpCond3; its use is determined at RESET initialization.
MPInvalidate	I	Multiprocessing Invalidate. Signals to the processor that it should issue invalidate data on the cache data bus. The address to be invalidated is externally provided. This is the same pin as CpCond2; its use is determined at RESET initialization.
$\overline{\text{Int}}$ (0-5)	I	A 6-bit bus used by the memory interface and coprocessors to signal maskable interrupts to the processor. At reset time, mode select values are read in.

2860 tbl 05

PIN DESCRIPTIONS (Continued)

Pin Name	I/O	Description
Clk2xSys	I	The master double frequency input clock used for generating $\overline{SysOut}$.
Clk2xSmp	I	A double frequency clock input used to determine the sample point for data coming into the processor and coprocessors.
Clk2xRd	I	A double frequency clock input used to determine the enable time of the cache RAMs.
Clk2xPhi	I	A double frequency clock input used to determine the position of the internal phases, phase1 and phase2.
$\overline{Reset}$	I	Synchronous initialization input used to force execution starting from the reset memory address. $\overline{Reset}$ must be deasseted synchronously but asserted asynchronously. The deassertion of $\overline{Reset}$ must be synchronized by the leading edge of $\overline{SysOut}$.

2860 tbl 06

ABSOLUTE MAXIMUM RATINGS(1, 3)

Symbol	Rating	Commercial	Military	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70(4) (Ambient) 0 to +90(5) (Case)	–55 to +125 (Case)	°C
TBIAS	Case Temperature Under Bias	–55 to +125(4) 0 to +90(5)	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
IIN	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

2860 tbl 07

NOTE:
1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. VIN minimum = -3.0V for pulse width less than 15ns. VIN should not exceed VCC +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.
4. 16-33 MHz only.
5. 37-40 MHz only.

AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V

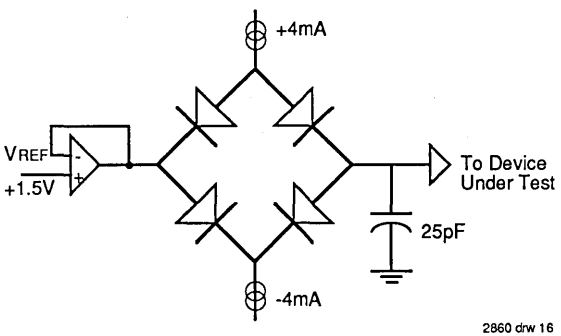
2860 tbl 08

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	Vcc
Military 16-33 MHZ	-55°C to +125°C (Case)	0V	5.0 ±10%
Commercial 16-33 MHz	0°C to +70°C (Ambient)	0V	5.0 ±5%
Commercial 37-40 MHz	0°C to +90°C (Case)	0V	5.0 ±5%

2860 tbl 09

OUTPUT LOADING FOR AC TESTING



2860 drw 16

DC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3000A				79R3000AE				Unit
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
C_{IN}	Input Capacitance(6)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	450	—	550	—	650	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	-100	100	-100	100	µA

2860 tbl 10

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag busses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

DC ELECTRICAL CHARACTERISTICS—
MILITARY TEMPERATURE RANGE (T_C = -55°C to +125°C, V_{CC} = +5.0V ±10%)

Symbol	Parameter	Test Conditions	79R3000A 16.67MHz Min.	79R3000A 16.67MHz Max.	79R3000A 20.0MHz Min.	79R3000A 20.0MHz Max.	79R3000AE 25.0MHz Min.	79R3000AE 25.0MHz Max.	79R3000AE 33.33MHz Min.	79R3000AE 33.33MHz Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
C_{IN}	Input Capacitance(6)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	500	—	600	—	650	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	-100	100	-100	100	µA

2860 tbl 11

5

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag busses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

DC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE (T_C = 0°C to +90°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3000AE				Unit
			37.0MHz		40.0MHz		
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(6)		—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	825	—	850	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	µA

2860 tbl 12

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag busses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

			79R3000A				79R3000AE				
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
Symbol	Parameter	Test Conditions	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Unit
Clock											
TCkHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TCkLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TCkP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		9	tcyc/4	7	tcyc/4	5	tcyc/4	3.5	tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	2	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type (2)	Load= 25pF	—	17	—	14	—	12	—	8.5	ns
TMWr	Memory Write	Load= 25pF	—	27	—	23	—	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
TAval	Address Valid	Load= 25pF	—	2	—	2	—	1.5	—	1	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		9	—	8	—	6	—	4.5	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	1	18	1	13.5	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	27	—	23	—	18	—	10	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSExc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	128	—	128	—	Tcyc
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	2	0.5	1	0.5	1	0	1	ns/25pF

2860 tbl 13

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33.33MHz; clock transition time < 5ns for other speeds.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— MILITARY TEMPERATURE RANGE (Tc = -55°C to +125°C, Vcc = +5.0V ±10%)

Symbol	Parameter	Test Conditions	79R3000A				79R3000AE				Unit
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Clock											
TCkHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TCkLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TCkP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		9	tcyc/4	7	tcyc/4	5	tcyc/4	3.5	tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	3	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type (2)	Load= 25pF	—	17	—	14	—	12	—	8.5	ns
TMWr	Memory Write	Load= 25pF	—	27	—	23	—	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
TAval	Address Valid	Load= 25pF	—	2	—	2	—	1.5	—	1	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		9	—	8	—	6	—	4.5	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	—	18	—	13.5	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	27	—	23	—	18	—	10	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSExc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	128	—	128	—	Tcyc
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	2	0.5	1	0.5	1	0	1	ns/25pF

2860 tbl 14

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33.33MHz; clock transition time < 5ns for other speeds.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— COMMERCIAL TEMPERATURE RANGE (Tc = 0°C to +90°C, Vcc = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3000AE				Unit
			37.0MHz		40.0MHz		
			Min.	Max.	Min.	Max.	
Clock							
TCkHigh	Input Clock High[2]	Note 7	5.5	—	5	—	ns
TCkLow	Input Clock Low[2]	Note 7	5.5	—	5	—	ns
TCkP	Input Clock Period[2]		13.5	500	12.5	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		3.5	tcyc/4	3	tcyc/4	ns
Run Operation							
TDEn	Data Enable[3]		—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	1.5	ns
TWrDly	Write Delay	Load= 25pF	—	2	—	2	ns
TDS	Data Set-up		4.5	—	4	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		6	—	6	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	3.5	—	3	ns
TAT2	Access Type (2)	Load= 25pF	—	8.5	—	7.5	ns
TMWr	Memory Write	Load= 25pF	—	9.5	—	9	ns
TExc	Exception	Load= 25pF	—	3.5	—	3	ns
TAval	Address Valid	Load= 25pF	—	1	—	1	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		4.5	—	4	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	ns
Stall Operation							
TSAVal	Address Valid	Load= 25pF	—	15	—	12.5	ns
TSAcTy	Access Type	Load= 25pF	—	13.5	—	9	ns
TMRdi	Memory Read Initiate	Load= 25pF	—	13.5	—	9	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	10	—	9	ns
TStl	Run Terminate	Load= 25pF	2	6.5	2	6	ns
TRun	Run Initiate	Load= 25pF	—	3	—	3	ns
TSMWr	Memory Write	Load= 25pF	2	9.5	2	9	ns
TSExc	Exception Valid	Load= 25pF	—	6.5	—	6	ns
Reset Initialization							
TRST	Reset Pulse Width		6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate[6]		0.5	1	0	1	ns/25pF

2860 tbl 15

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns.

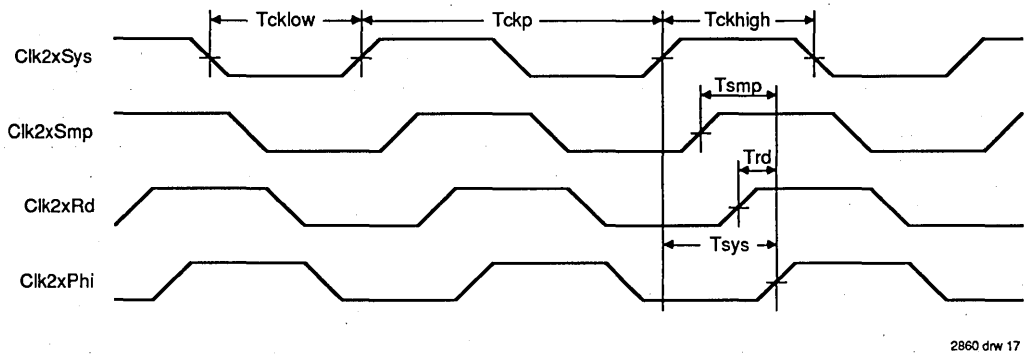


Figure 12. Input Clock Timing

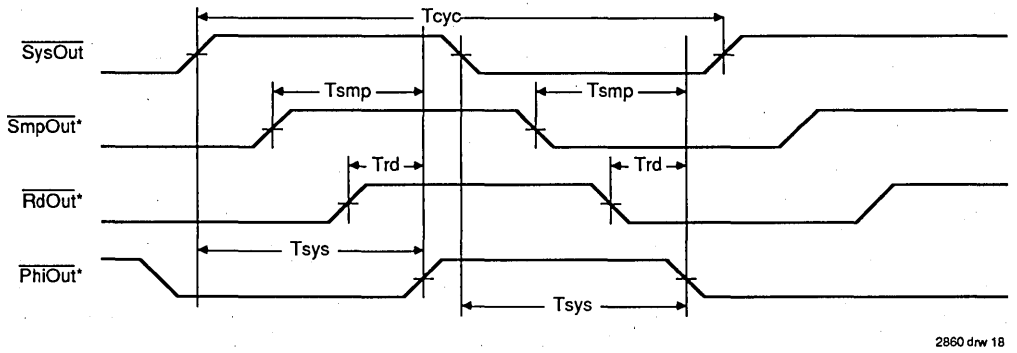


Figure 13. Processor Reference Clock Timing

* These signals are not actually output from the processor.
They are drawn to provide a reference for other timing diagrams.

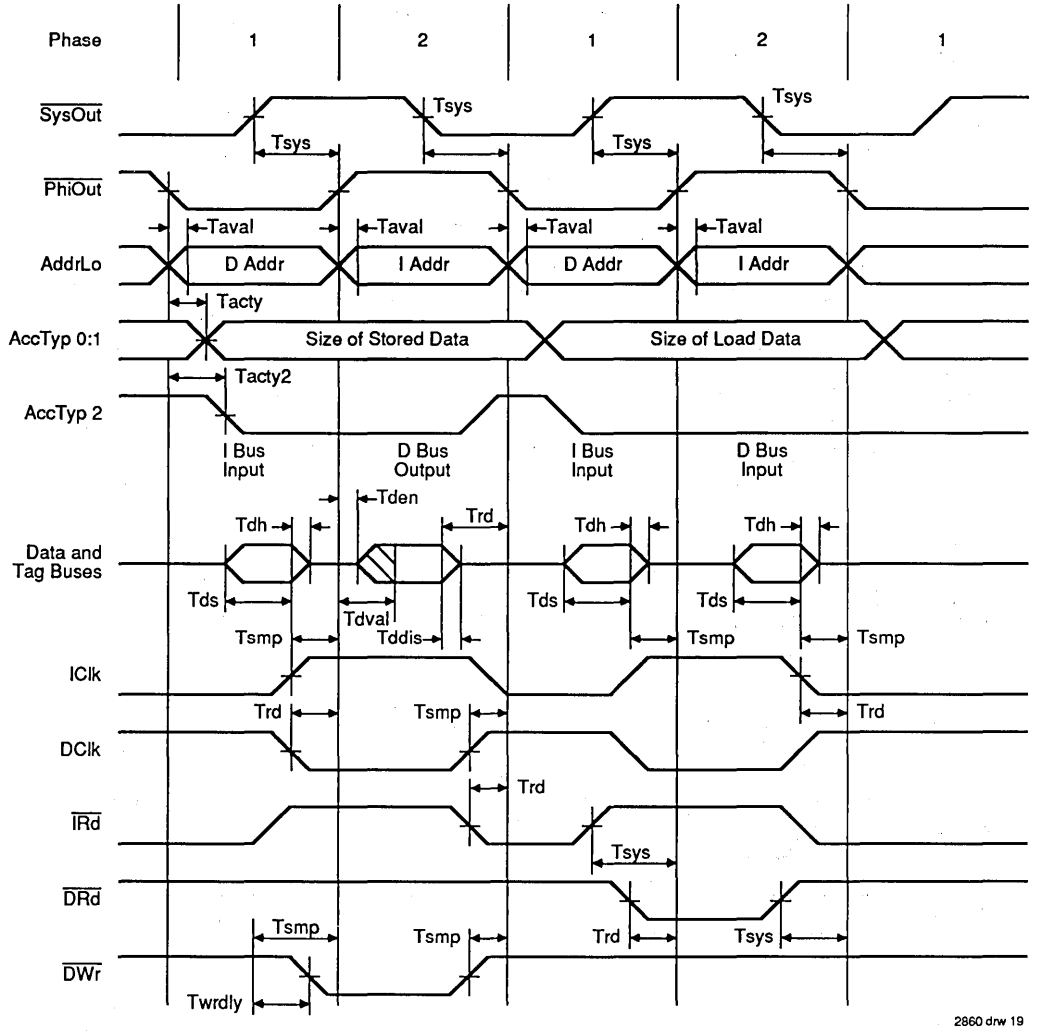


Figure 14. Synchronous Memory (Cache) Timing

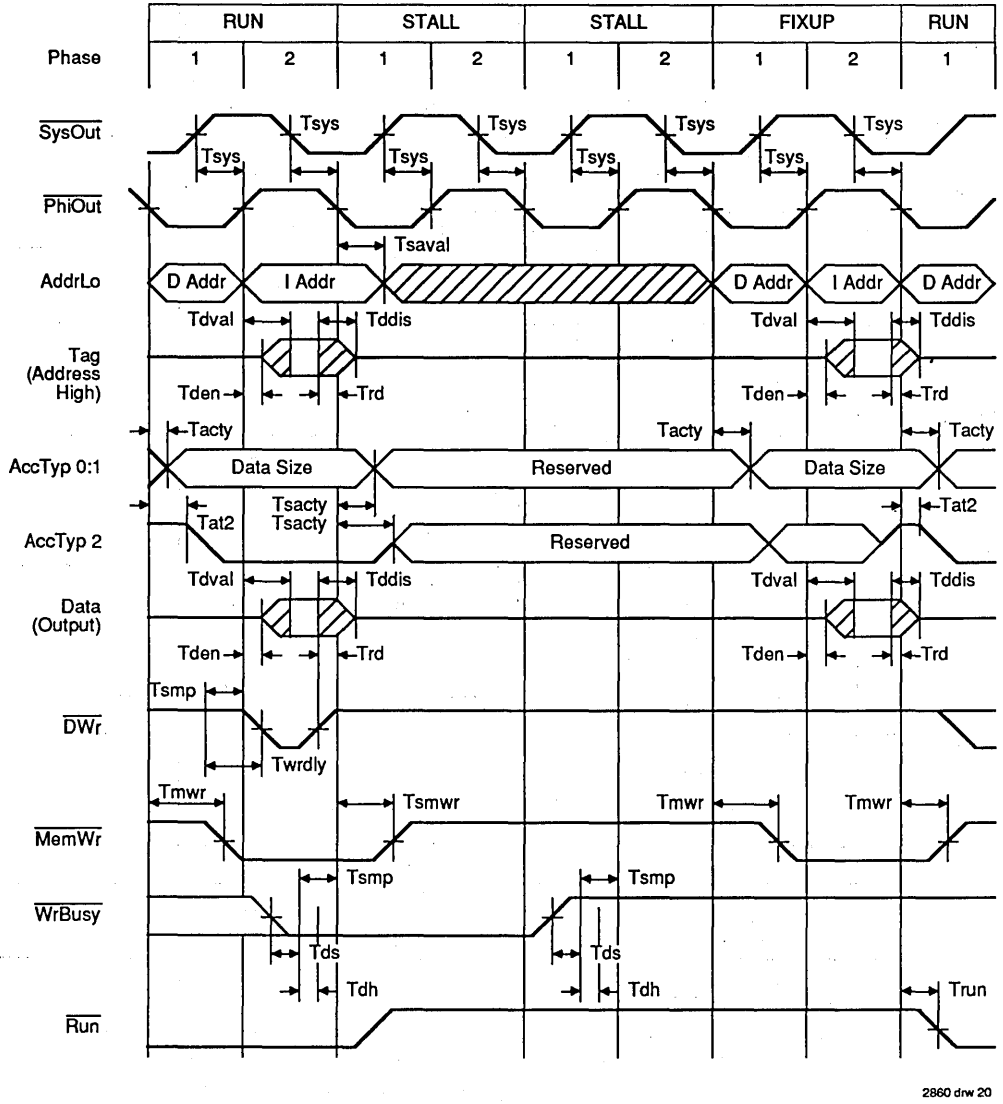


Figure 15. Memory Write Timing

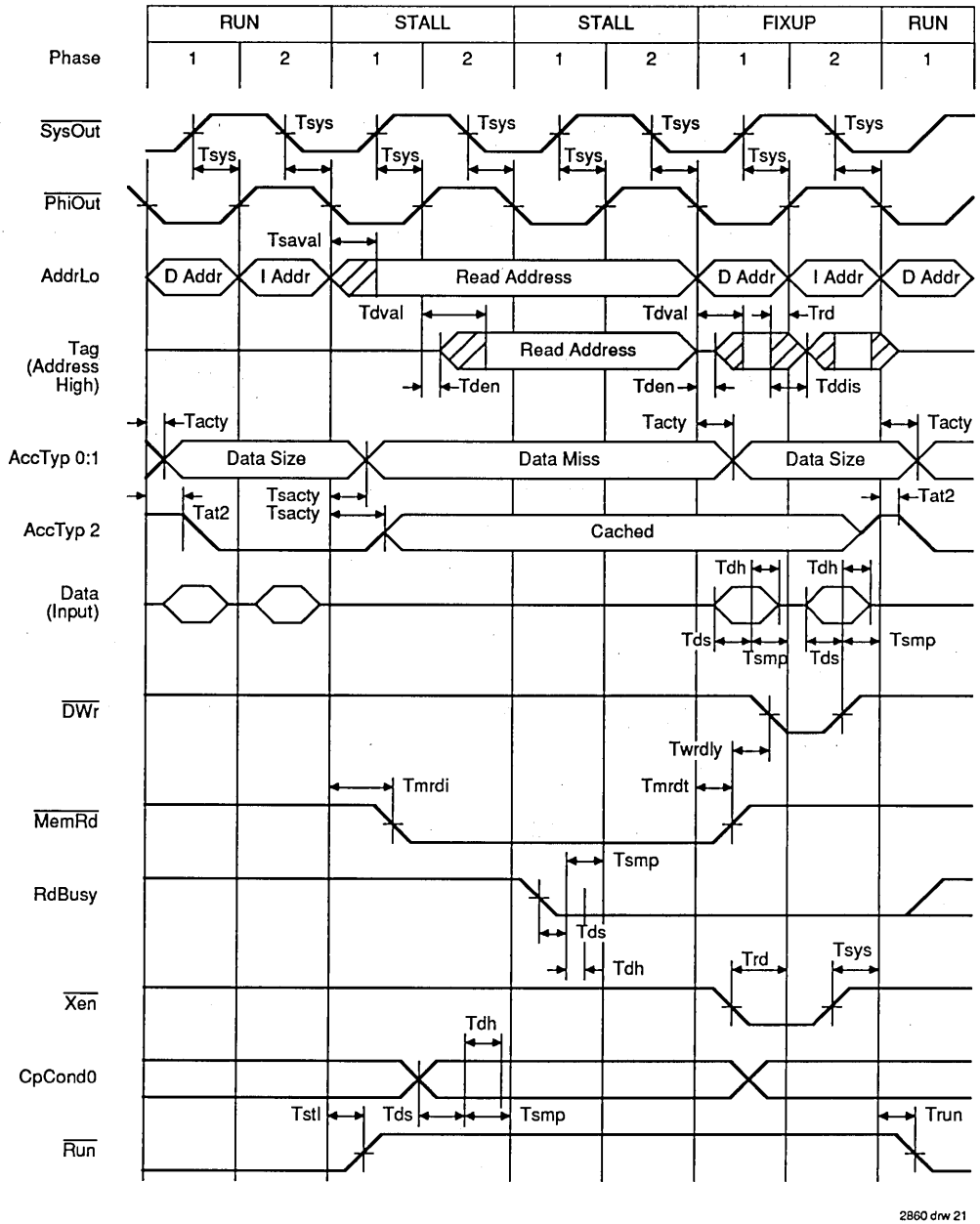


Figure 16. Memory Read Timing

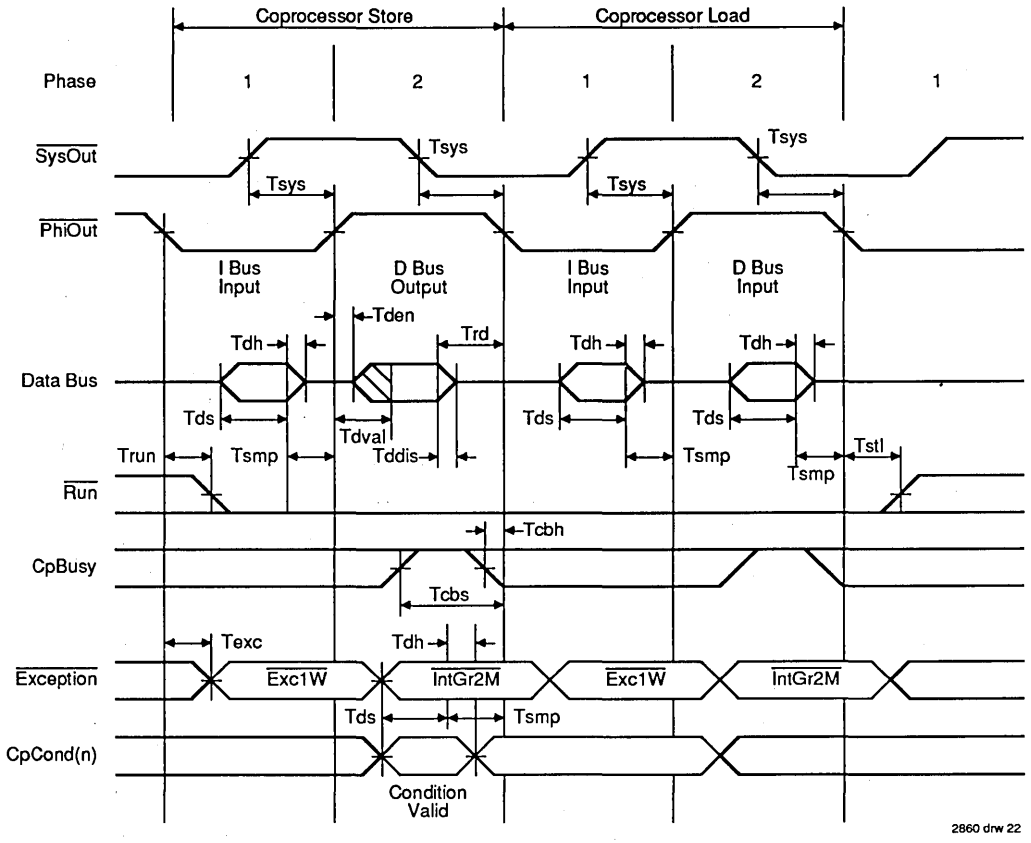


Figure 17. Coprocessor Load/Store Timing

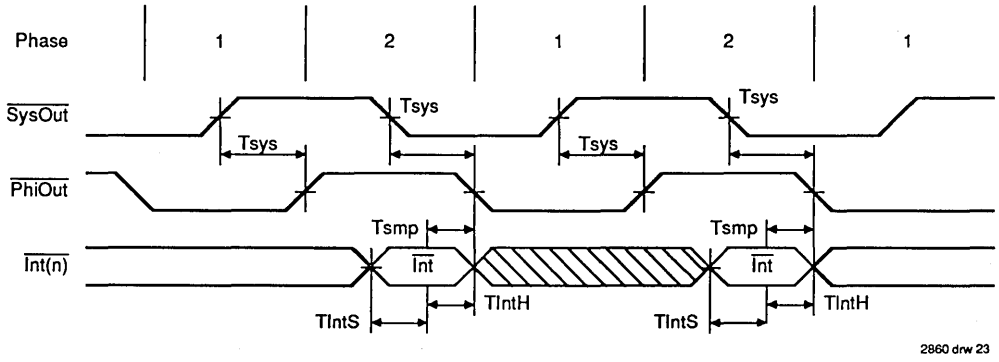


Figure 18. Interrupt Timing

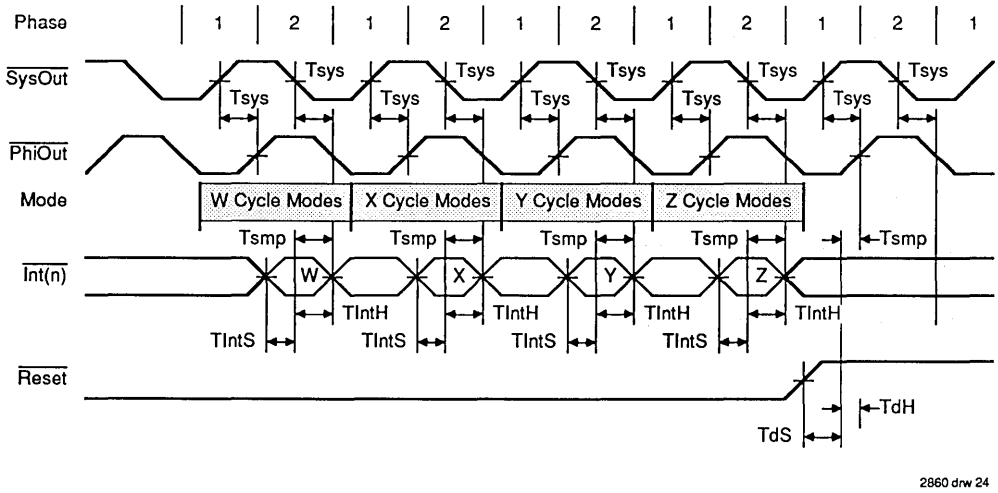
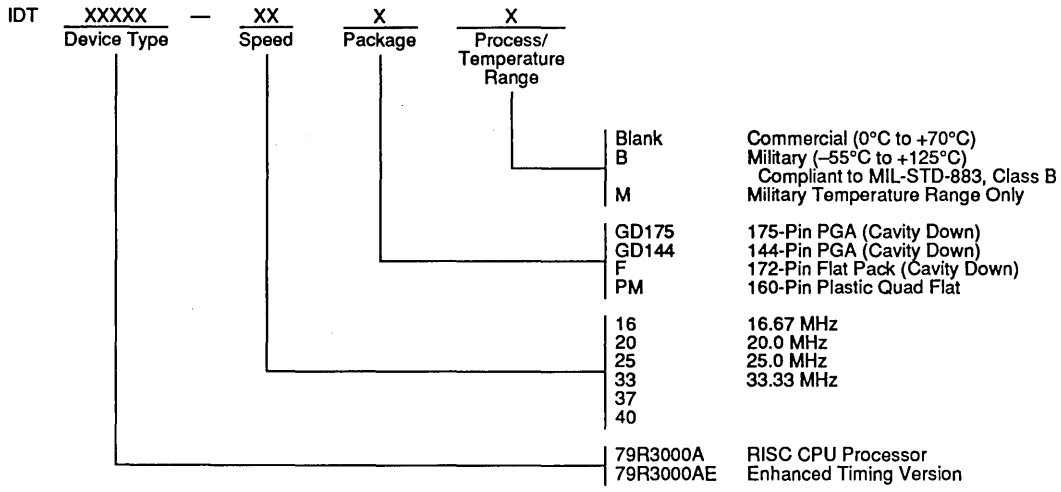


Figure 19. Mode Vector Initialization

NOTES:
1. Reset must be negated synchronously; however, it should be asserted asynchronously. Designs must not rely on the proper functioning of SysOut prior to the assertion of Reset.
2. If Phase-Lock On or R3000 Mode are asserted as mode select options, they should be asserted throughout the Reset period, to insure that the slowest coprocessor in the system has sufficient time to lock to the CPU clocks.
3. Reset is actually sampled in both Phase 1 and Phase 2. To insure proper initialization, it must be negated relative to the end of Phase 1.

ORDERING INFORMATION



2860 drw 25

VALID COMBINATIONS

IDT	79R3000A - 16, 20	All packages
	79R3000A - 16, 20B, M	GD144, GD 175, F
	79R3000AE - 25, 33 B, M	GD144, GD 175, F
	79R3000AE - 25, 33, 37, 40	GD144, GD 175, F

AE2860-0

Integrated Device Technology, Inc.

RISController™ CPU FOR HIGH-PERFORMANCE EMBEDDED SYSTEMS

IDT79R3001

FEATURES:

- Enhanced Instruction Set compatible version of IDT79R3000 RISC CPU
- Achieves high-performance with reduced parts count and lower overall system cost
- Flexible on-chip cache controller supports various cache, main memory sizes
- Supports optional data parity with parity error output signal
- Works with IDT79R3010 RISC Floating-Point Coprocessor
- DMA interface support
- Large synchronous memory space for real-time systems
- Full 32-bit operations — 32-bit registers, 32-bit address and data interface
- On-chip memory management unit with 64 fully associative TLB entries maps 4 Gbyte virtual address space
- High-speed interrupt response (6 interrupt input pins) with precise exception capability
- High-speed CEMOS™ technology results in speeds from 12.5 to 40MHz
- Supports caches from 8 Kbytes to 16Mbytes
- Independent block refill sizes for the instruction and data caches
- Concurrent cache refill and execution
- Works on 8-, 16- and 32-bit data
- Supports unaligned 32-bit data
- Optimizing compilers for C, Ada, Pascal, Fortran
- RTOS support for C or Ada environments

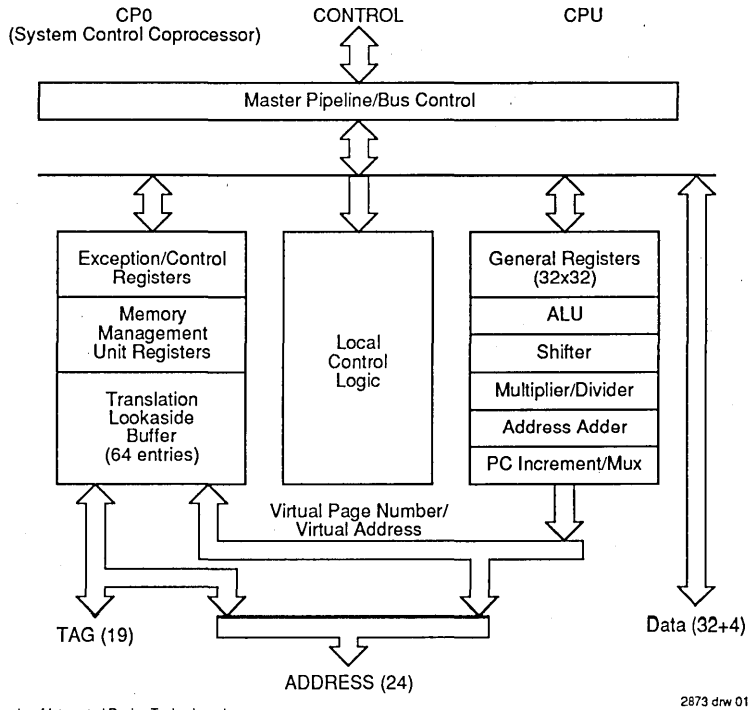
DESCRIPTION:

The IDT79R3001 brings the high-performance inherent in the IDT79R3000 RISC Microprocessor to lower cost systems. It does this while maintaining full (both User and Kernel) software compatibility with both the IDT79R2000A and IDT79R3000 RISC Microprocessors.

The IDT79R3001 achieves lower system cost by reducing the number of components required to construct a synchronous memory (or cache) external to the processor and by simplifying the asynchronous memory interface. By removing the requirement for parity and allowing the system designer to select the cache organization which best suits the system,

5

FUNCTIONAL BLOCK DIAGRAM



CEMOS and RISController are trademarks of Integrated Device Technology, Inc.

overall parts count is dramatically reduced while maintaining high performance.

The IDT79R3001 RISC Microprocessor extends the ability of the IDT79R3000 family to support embedded and cost sensitive applications. Its level of integration and flexibility allows high-performance systems to be constructed at reasonable cost in a straightforward manner, without forcing the system designer to support features not required in his application.

The IDT79R3001 consists of two tightly coupled processors integrated on a single chip. The first processor is a full 32-bit CPU based on RISC principles to achieve a new standard of performance in microprocessor based systems. The second processor is a system control co-processor, called CP0, containing a fully associative 64-entry TLB (Translation Lookaside Buffer), MMU (Memory Management Unit), and control registers, supporting a 4 Gigabyte virtual memory subsystem and a Harvard Architecture Synchronous Memory/ Cache controller which achieves ultra-high bandwidth using industry standard SRAM devices.

This data sheet provides an overview of the features and architecture of the IDT79R3001 CPU. A more detailed description of the operation and timing of this device is incorporated in the "IDT79R3001 Hardware User's Guide," and a detailed architectural overview is provided in the "mips RISC Architecture" book, both available from IDT. Further literature describing the hardware, software, and development tools for the IDT79R3001 are also available from IDT.

HARDWARE OVERVIEW

The IDT79R3001 is a high-performance RISC microprocessor incorporating a fast execution engine and sophisticated yet flexible memory interface designed to support the processor bandwidth requirements at minimal system cost.

Execution Engine

The IDT79R3001 contains the same basic execution engine as the ultra-high performance IDT79R3000 and thus achieves over 28 MIPS performance at 33 MHz.

The key to the performance of the processor is the instruction pipeline, illustrated in Figure 2. The execution of a single IDT79R3001 instruction consists of five primary steps, some of which may be broken down further into smaller subsets.

The five primary stages of the pipeline, each of which require approximately one CPU cycle, are:

IF Instruction Fetch, when the processor fetches the instruction from the Instruction Synchronous Memory.

RD Read required operands from on-chip register file while decoding the instruction.

ALU Perform the required operation on instruction operands.

MEM Access data memory (load or store).

WB Write results back to register file.

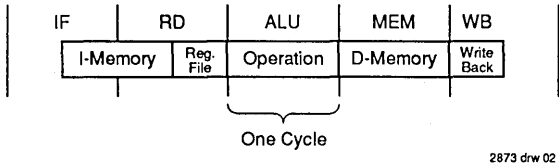


Figure 2. IDT79R3001 Five-Stage Pipeline

Thus, the CPU achieves an average execution rate approaching one instruction per CPU cycle, since the execution of five instructions at a time are overlapped within the processor (Figure 3). Optimizing compiler technology fully comprehends the interaction of software with the various pipeline resources, and serves to both eliminate any potential pipeline conflicts which might arise and to maximize instruction throughput.

The IDT79R3001 Memory Interfaces

The key to achieving the inherent performance of the IDT79R3001 is to design a memory subsystem capable of providing a new instruction to the processor on almost every clock cycle.

Like the IDT79R3000, the IDT79R3001 supports a hierarchical view of the memory subsystem. However, the IDT79R3001 allows the system designer to make more tradeoffs in the partitioning and architecture of the various levels in order to more completely meet the needs of certain types of applications.

The IDT79R3001 supports two classifications of external memory: synchronous and asynchronous. The Harvard-Architecture (separate instruction and data memories) synchronous memory allows the processor to achieve the highest levels of performance. The processor is able to obtain both an instruction and data word from the synchronous memory on every clock cycle, resulting in high instruction and data throughput.

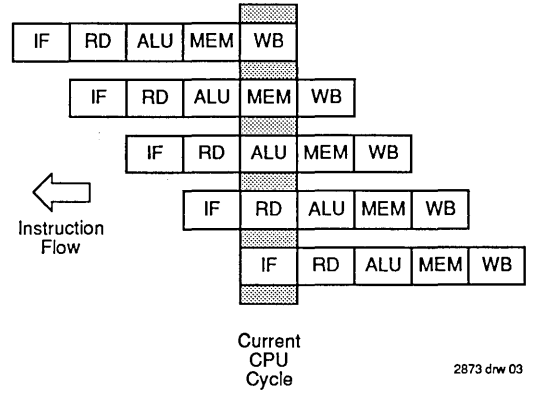


Figure 3. Instruction Execution in IDT79R3001 Pipeline

The asynchronous memory space contains larger, slower memory devices such as EPROM, main memory DRAMs, and peripheral devices. Multiple clock cycles are required for data movement in the asynchronous memory.

Many systems implement a memory hierarchy between these two memory spaces, whereby the synchronous memory space is used as processor caches and the asynchronous memory space is used for main memory. The IDT79R3001 integrates a flexible Direct-Mapped Cache Controller On-Chip, eliminating external cache control logic and minimizing cache management overhead. If the synchronous memory space is used for processor caches, then cache "misses" will cause the processor to automatically process an asynchronous memory transfer to refill the cache.

The key to achieving the system cost and performance goals of an IDT79R3001-based system is to partition the memory system to the needs of the application.

Synchronous Memory System

As with any high-performance processor, the IDT79R3001 requires high-bandwidth to achieve high-performance. Thus, it is important that the majority of its execution occur in the synchronous memory space. In applications which require substantial amounts of main memory, this memory space will be implemented as instruction and data caches.

The synchronous memory is designed to be able to supply both an instruction and data word to the processor on each clock cycle. When the synchronous memory spaces are used as caches, then they are used to hold instruction and data that is repetitively accessed by the CPU (for example, within a program loop). This reduces the number of slower asynchronous memory cycles and thus achieves higher performance.

Some microprocessors incorporate small amounts of cache on-chip, which has a very small and unpredictable effect on the execution of large programs. The IDT79R3001 supports

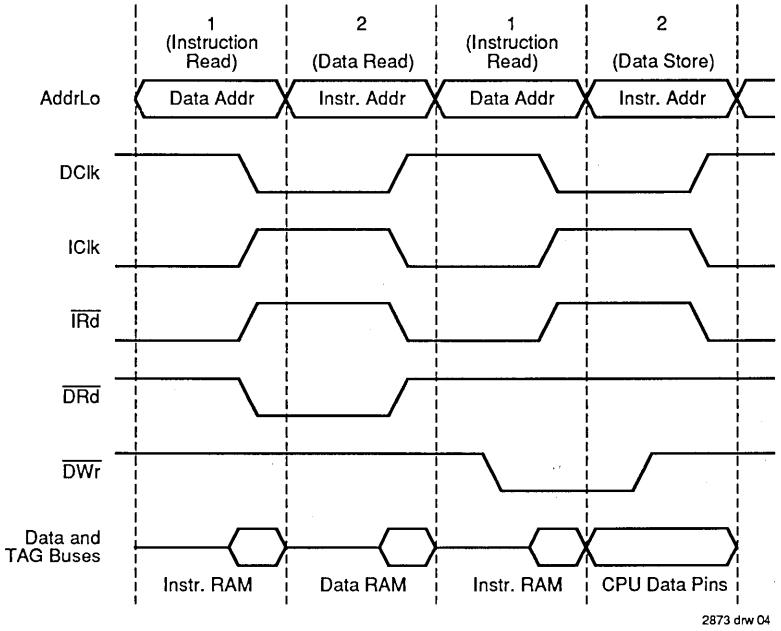


Figure 4. Synchronous Memory Control Timing

caches of from 8kB in size up through 16MB, thus bringing substantial performance improvements to very large programs and also allowing real-time system designers to design cache-based systems to support deterministic requirements.

The IDT79R3001 directly controls the synchronous memory interface (whether it is being used as caches or not) with a minimum of external components. The IDT79R3001 includes all control signals and cache TAG control logic (for a direct mapped cache) for the synchronous memory interfaces. Parity over the data portion of each synchronous memory can be optionally selected at RESET time for applications which desire to make this cost trade-off.

The synchronous interface works by dividing the basic CPU cycles into two phases. During one phase, a cache address is presented by the processor and captured by external latches (the latch control signals are directly generated by the CPU). During the next phase, the address for the other memory space is generated and captured while the data movement operation or the first cache is completed. The processor directly generates the SRAM Output Enable and Write Enable signals and the address latch enable signals, requiring no external decoding. This is illustrated in Figure 4.

Further, the IDT79R3001 supports the ability to refill multiple words into the cache from main memory when a cache-miss

occurs, further reducing system cost and increasing performance in cache-based systems. The IDT79R3001 can obtain 1, 4, 8, 16, or 32 words from main memory when processing a cache-miss, thus amortizing the cache-miss penalty over a large amount of data.

The IDT79R3001 also performs instruction streaming, which is the simultaneous execution of incoming instructions while the cache is being refilled.

The actual width of the tag bus, and whether or not parity over the data parts of each synchronous memory is included, is determined according to how the device is initialized. The IDT79R3001 can accommodate a TAG bus width of 0-19 bits, compatible with a variety of cache sizes and cacheable main memory choices. The IDT79R3001 allows the system designer to scale the synchronous memory system exactly according to the system needs, thus eliminating extra memory and logic devices and achieving substantial cost savings with no loss of performance.

Thus, the synchronous memory interface of the IDT79R3001 allows for high-bandwidth memory systems to be implemented with a minimum of control logic. This is desirable, since RISC performance tends to be a function of memory bandwidth. By simplifying the design of the synchronous memory system (illustrated in Figure 5), it is easier for the system designer to achieve high performance with minimum chip count and without requiring ultra-fast or specialty components.

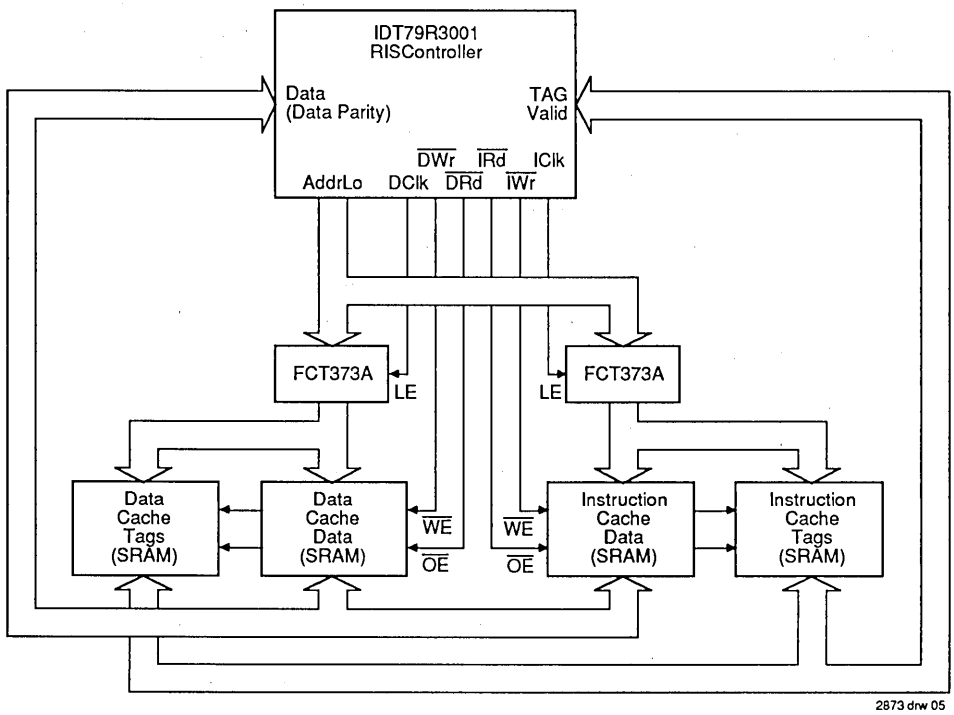


Figure 5. IDT79R3001 Synchronous Interface

The TAG Bus

The TAG bus of the IDT79R3001 has been designed to allow the system designer to implement the exact cache configuration that is right for the system. For larger caches, low-order TAG bits do not need to be supplied for the TAG comparison. Additionally, the number of high-order TAG bits supplied is determined by the system designer, according to the amount of cacheable main memory the system supports. Since most embedded systems would tend to implement caches of 16KB and greater, and cacheable memory spaces of 32MB or smaller, significant cost and area reductions are achieved by configuring a smaller TAG bus.

The system configures the on-chip TAG comparator at RESET Initialization time. If a TAG bit is not to be included in the synchronous memory TAG bit compare, a pull-down resistor of 4kΩ is connected to the appropriate IDT79R3001 TAG pin. If a TAG bit is to be included, no resistor is required (the IDT79R3001 pulls floating inputs to Vcc during RESET by a small pull-up, which is disabled when RESET is negated).

If a TAG bit is excluded from the cycle-by-cycle comparison, it is still driven out with the appropriate address value during write cycles or asynchronous memory reads. Thus, the system designer still has the full 4 Gbyte of address space available for address decoding, without requiring the synchronous memory to be able to cache all such addresses.

Figure 6 illustrates a reduced system, which implements 16KB of Instruction and 16KB of data cache, and 512MB of cacheable address space, using just 6 IDT71586 4Kx16 Latched CacheRAM™ components and 4 pull-down resistors.

Note that in systems which do not implement the synchronous memory space as cache, then pull-down resistors would be added to all TAG pins. The Valid Pin still needs to be supplied on each cycle, thus allowing various memory schemes to be implemented (such as static column DRAM). However, the IDT79R3001 can be initialized to not assert the Valid pin as an output during Write cycles, simplifying the design of logic to drive the signal.

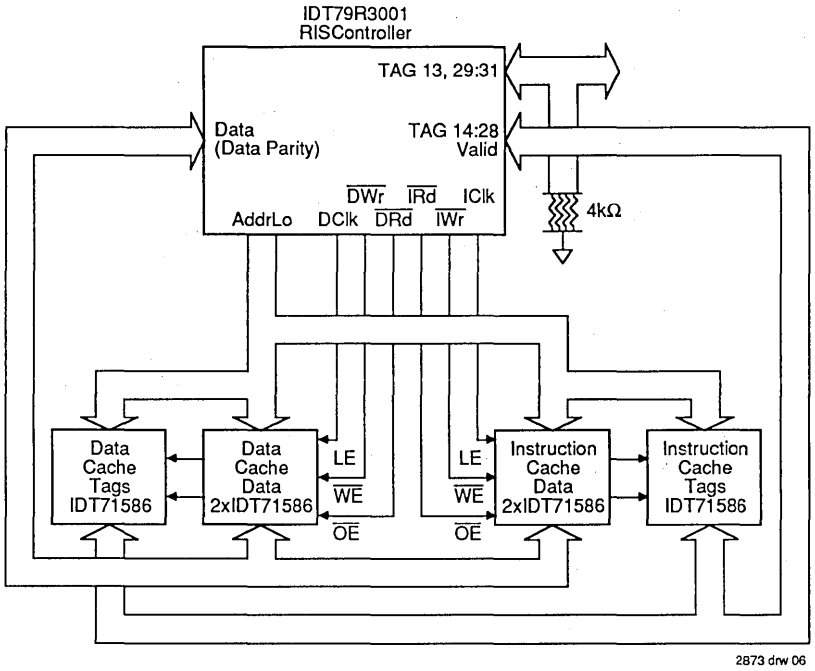


Figure 6. Small Footprint Cache for IDT79R3001

Cache Update

When the on-chip TAG comparator indicates that the item read from the cache was not the desired item, a cache-miss is processed. A main memory (asynchronous) transfer is automatically processed.

The IDT79R3001 desires to update the cache using a burst refill of multiple adjacent words from main memory. The processor is "stalled" until the first word of the block is available. The processor is then released, and the block of words is brought into the cache at the rate of one word per CPU clock cycle.

Note that if the cache-miss was in the instruction cache, the processor is capable of simultaneously executing the incoming instruction stream as the cache is updated, thus effectively making the cache update transparent to the system and increasing performance.

Write Cycles

The IDT79R3001 utilizes a write through cache. That is, data written by the processor is both written to the cache and main memory simultaneously. Thus, main memory always has a current copy of all data.

Typically, latching devices are used between the cache subsystem and the slower main memory. These Write Buffers capture the data simultaneous with the cache update, allowing the processor to continue to the next cycle without actually waiting for the main memory transfer to complete. The IDT79R3001 generates parity over the data field on write cycles, which can be propagated into both the synchronous and asynchronous memory spaces.

When the processor writes less than a 32-bit quantity (a "partial" word), the processor can perform a "read-modify-write" of the cache. That is, the processor will read the 32-bit word containing the partial address(es) to be updated from the cache. If a "hit" occurs, then the new data will be merged with the old and the new 32-bit value will be written both to the cache and to main memory. If a cache "miss" occurs, then only the partial data is written to main memory and the cache is unchanged. Partial word capability is selected as a RESET option.

THE ASYNCHRONOUS MEMORY INTERFACE

The IDT79R3001 also supports an asynchronous memory interface, which supports the use of slower memory devices such as slow DRAM or EPROM and also supports the use of peripherals and other "non-cacheable" devices.

In general, if a cache-miss (or parity error, if enabled) occurs, the processor will automatically use the asynchronous memory interface to retrieve the desired data, and will update the cache accordingly.

Additionally, software can force the use of the asynchronous memory space through the use of the on-chip MMU. When the processor seeks either instructions or data within a certain address range (kseg1), the processor knows that this data is uncacheable and will perform an asynchronous memory transfer. Additionally, within cacheable memory, TLB entries can be used to make certain pages as "uncacheable". When an address of an "uncacheable" page is used, the processor will automatically use the asynchronous memory space.

The asynchronous memory space uses the same data bus as the synchronous memory space. This facilitates the automatic updating of cache memory when the asynchronous memory is accessed due to cache-miss activity or memory writes. The asynchronous address bus is composed from the synchronous memory AddrLo bus, and the TAG bus. External logic devices (such as IDT74FCT374A registers) are used to capture AddrLo and TAG values for the asynchronous transfer address. Note that systems which exclude individual TAG bits from comparison (to reduce cache width) still have all TAGs available as outputs.

The data path between the processor and the asynchronous memory space is managed according to the needs of the application. Write Buffer FIFO devices, such as the IDT79R3020, are used to capture address and data during store cycles. These devices are used to capture the data in one-cycle, and allow the processor to continue to execute from the synchronous memory while the slower asynchronous memory actual retires the write.

The read path is also constructed according to the needs of the system. If block refill is used, then the read path is highly dependent on the design of the main memory system. Pipeline devices such as IDT74FCT540A, or simple latches such as IDT74FCT374, may be used.

A simple asynchronous memory interface is shown in Figure 7. In this system, main memory is assumed to be fast enough to support the block refill requirements of the system, thus simplifying the read path. In fact, both the read and write data paths are actually managed through a single set of IDT29FCT52A bidirectional latching transceivers.

During write cycles (which are typically captured by Write Buffers), the processor asserts $\overline{\text{MemWr}}$ to indicate that a write cycle is in progress. The memory system negates WrBusy to indicate that the processor is done with the write cycle.

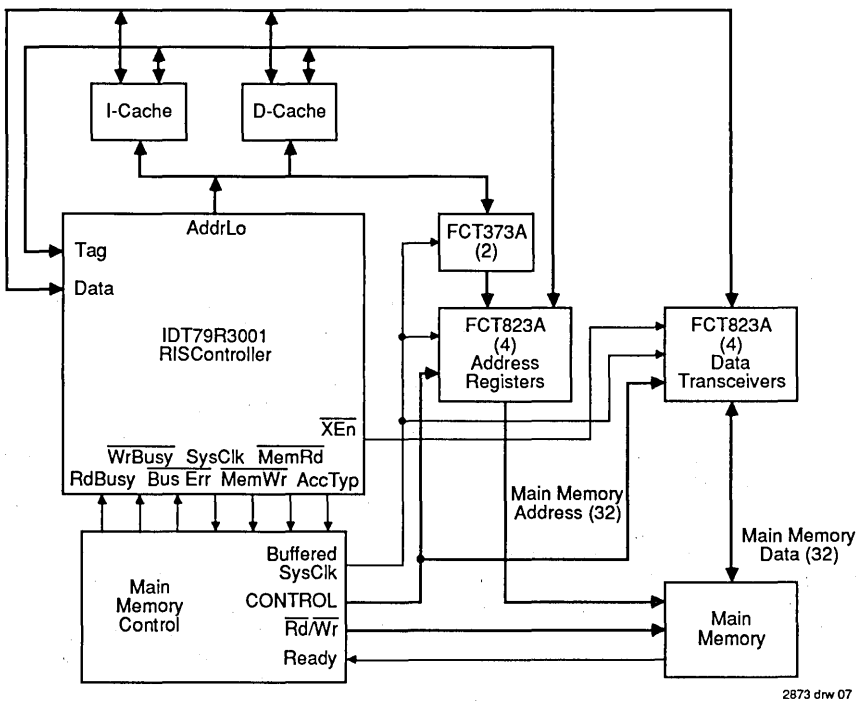


Figure 7. IDT79R3001 Asynchronous Interface

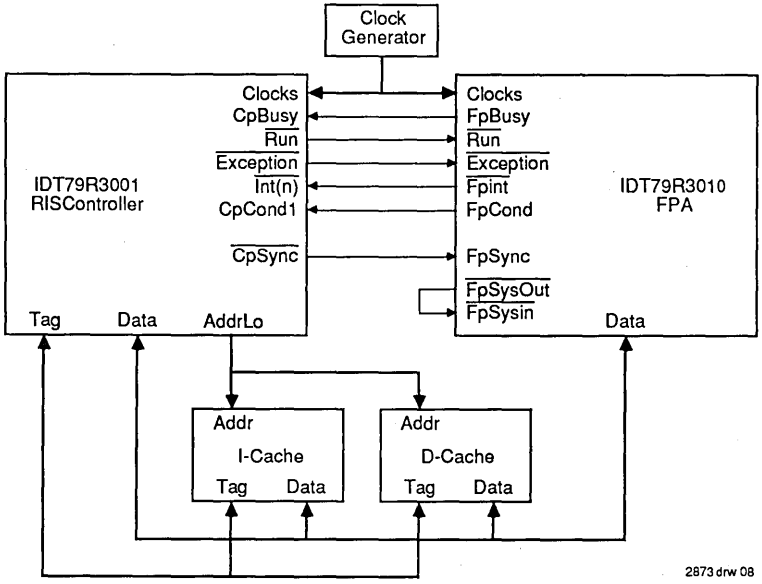


Figure 8. IDT79R3001 Interface to IDT79R3010 Floating Point Co-Processor

During read cycles, the processor will assert $\overline{\text{MemRd}}$ to indicate that a main memory read is in progress. The memory system will hold RdBusy active until the desired data is available. The processor will activate the $\overline{\text{XEn}}$ signal to allow data to be passed from the main memory to the processor databus. If the cache is to be updated with the new data, then the processor will assert the appropriate cache write signal to allow the cache RAMs to capture the incoming databus.

The AccTyp bus is used to indicate the size of the data transfer (8, 16, 24, or 32 bits), and for main memory reads, whether or not the data is "cacheable". This simplifies the main memory address decoding, since the AccTyp indicates whether the main memory needs to perform a burst read of multiple words.

Co-Processor Interface

The IDT79R3001 implements a co-processor interface, which allows the use of the IDT79R3010 high-performance RISC Floating Point Accelerator without requiring the use of external interface components.

The co-processor interface has been designed to make system co-processors appear to the programmer as if they were on-chip extensions of the core execution engine. Thus, the IDT79R3010 FPA works as a true co-processor, rather than as a peripheral which must be programmed.

In the IDT79R3001 co-processor model, the CPU is responsible for controlling all data cycles. The co-processor keeps in synchronization with the CPU (including the pipeline stages), and uses a Phase-Locked Loop to keep synchronized with the processor bus traffic. The co-processor then "snoops" the data bus, watching for co-processor instructions. It also knows when data cycles on the bus are intended for it (either as a target in co-processor load operations, or as a source for co-processor restore operations), and performs the data portion of the operation when appropriate. Thus, co-processors effectively load and store directly with memory, without requiring operands to go through the CPU first. This achieves the highest levels of performance (note that the co-processor interface also supports move, whereby data can be moved directly between the CPU and any co-processor).

Figure 8 illustrates the use of the IDT79R3010 in a IDT79R3001 system. The co-processor interface manages synchronization between the parts, and is used to communicate status from the co-processor to the CPU. CpBusy, or co-processor busy, stalls the CPU until the busy co-processor resource (requested by a co-processor instruction) is free, and CpCond, or co-processor condition, is used to report status on co-processor test instructions. CpSync, is used to help the co-processor stay "locked" to the CPU, so that the co-processor knows when data is on the bus to be sampled on load operations or when to place data on the bus for store operations.

Note that the co-processor sits on the same data bus as the CPU, but has no connection to the address bus. The CPU is responsible for performing all memory addressing, including the determination of "cache hit", write-buffer full cycles, and any processing that might be required for cache misses.

INTERRUPTS

The IDT79R3001 features 6 separate interrupt input pins. Interrupts are not vectored, but rather cause the general exception vector address to be the next execution address.

These pins are not encoded internally; external logic can choose to implement these interrupt lines as either 6 or 64 interrupt sources; software would then perform the appropriate decoding to get to the specific interrupt handler.

Interrupts are recognized in the ALU stage of the on-chip pipeline. Instructions less advanced in the pipeline are "flushed" and will be restarted when the return from exception occurs (an on-chip register contains the address of the instruction which was excepted). Instructions further advanced in the pipeline are allowed to continue. Unlike other RISC processors, the IDT79R3001 does not require the programmer to save and restore pipeline status to allow normal execution to be resumed. Depending on the application and exception, at most software would need to save/restore the on-chip data registers, status register, Exception PC and exception "cause" register.

Note that the co-processor model includes "precise exceptions." That is, an exception is signaled to the exact instruction which generated the exceptional condition. No further state commitments are made by the IDT79R3001 and, thus, the exact context at the time of the exception is known to the programmer. This is true even for multi-cycle operations, such as those of the FPA.

DMA INTERFACE

The IDT79R3001 features a simple DMA interface which allows an external master to gain control of the synchronous memory space. Note that it is not necessary to include logic on the CPU to arbitrate for the asynchronous memory space; the read/write buffer interface is where such arbitration logic belongs and it is left to the system designer to implement the type of asynchronous memory structure that best fits the application.

When an external master "owns" the synchronous bus, the CPU will tri-state the following pins and buses:

AddrLo: The Synchronous memory direct address bus.

Data & Tag: The synchronous memory RAM data lines.

Cache Control: $\overline{IRd}$, $\overline{IWr}$, ICIk, $\overline{DRd}$, $\overline{DWr}$ and DCIk. This allows the external master to use the existing control lines to control the synchronous memory.

XEn: The read buffer transceiver enable, which will allow the external master to use the read/write buffer path for DMA.

Valid: This enables the DMA interface to be used for multi-processing applications.

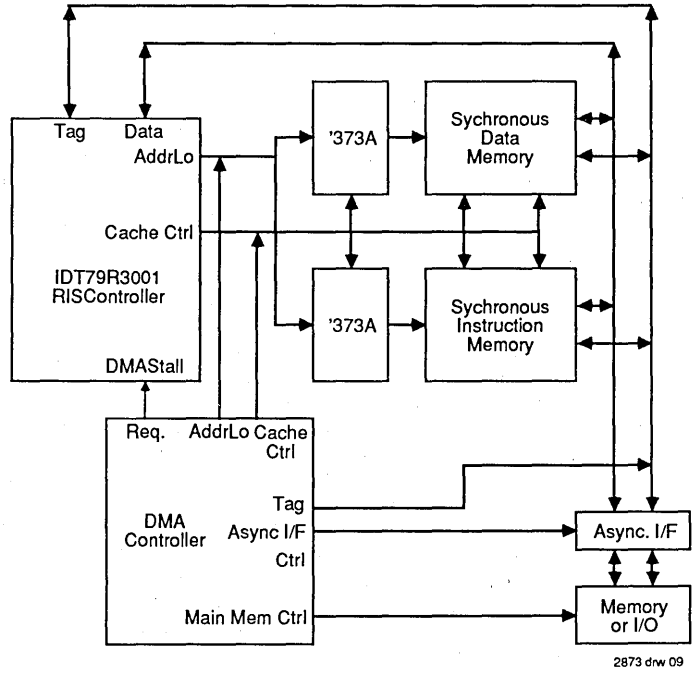


Figure 9. IDT79R3001 DMA Interface

Input	W Cycle	X Cycle	Y Cycle	Z Cycle
$\overline{Int0}$	Reserved	Reserved	Reserved	Reserved
$\overline{Int1}$	Reserved	Reserved	Reserved	Reserved
$\overline{Int2}$	$\overline{DBlkSize0}$	$\overline{DBlkSize1}$	$\overline{Parity\ On}$	Valid Output
$\overline{Int3}$	$\overline{IBlkSize0}$	$\overline{IBlkSize1}$	StorePartial	ControlLow
$\overline{Int4}$	$\overline{PllOn}$	$\overline{PllOn}$	$\overline{PllOn}$	$\overline{PllOn}$
$\overline{Int5}$	Reserved	BigEndian	$\overline{TriState}$	Reserved

NOTE:
1. Reserved signals must be "high" during these cycles.

2873 tbl 01

Table 1. IDT79R3001 Mode Selectable Features

The DMA interface consists of a single input signal, DMAStall, which causes the processor to stall and to tri-state the above named lines. The external master is guaranteed mastership of the bus within a very short number of cycles, depending on the exact external bus activity of the CPU when the DMA was requested. The DMA master negates the DMAStall signal when the DMA operation is completed to allow the CPU to resume processing. Consult the "IDT79R3001 Hardware User's Guide" for more details.

Figure 9 illustrates the system connection of an external DMA master to a IDT79R3001 system.

ADVANCED FEATURES

The IDT79R3001 contains special features which provide added flexibility across a number of applications, as well as allow for system diagnostic support.

In support of diagnostics, the IDT79R3001 allows for cache "swapping" (interchange of which memory bank is for instruction and which is for data), which is useful in system initialization, cache flushing, and diagnostics. Additionally, the caches can be "isolated" from main memory, which forces cache "hits" to occur regardless of the tag comparison, and which is useful in determining that the synchronous memory space RAMs are functional.

An additional feature is the ability to enable parity checking over the data field of each synchronous memory. If parity is enabled, the processor will check the parity when a synchronous access occurs; if a parity error is detected, it is signaled to the external world on the $\overline{Parity\ Error}$ signal and a cache-miss cycle is processed. the $\overline{Parity\ Error}$ signal will remain low until the parity error flag in the CP0 status register is cleared by software.

A number of other system selectable features are selected at reset time. The input reset "vectors" are sampled on the interrupt input lines during the last four cycles of the reset period. The input vectors are listed in Table 1. These selections include the ability to select the block refill sizes for each of the instruction and data memories, whether Big Endian or Little Endian order is to be used, whether to use data parity, and whether or not to accommodate a Phase-Locked Loop for a co-processor. The initialization of the CPU and meaning of each input vector is more fully explained in the "IDT79R3001 Hardware User's Guide".

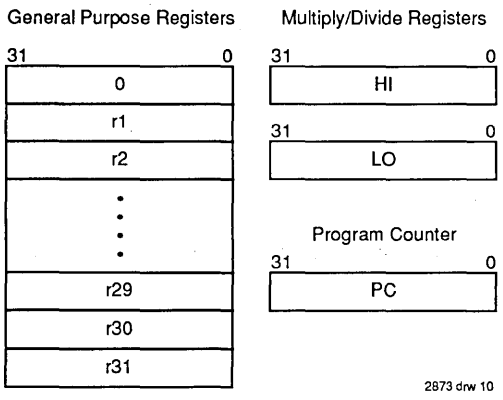


Figure 10. IDT79R3001 Registers

PROCESSOR ARCHITECTURE

The IDT79R3001 is a full implementation of the IDT79R2000A/IDT79R3000 Instruction Set Architecture (the MIPS-I ISA). This architecture is discussed in great detail in "mips RISC Architecture," available from IDT.

IDT79R3001 CPU Registers

The IDT79R3001 CPU provides 32 general purpose (orthogonal) 32-bit registers, a 32-bit Program Counter and two 32-bit registers used to hold the results of the CPU integer multiply and divide operations.

Two of the 32 general registers have special purposes designed to increase processor performance: register r0 is hardwired to the value "0", a useful constant; and register r31 is used as the link register in jump-and-link instructions (the return address for subroutine calls). Otherwise, there is no requirement that a particular register be used as a stack or frame pointer, etc., although there is a register convention as part of the "mips ABI" (Applications Binary Interface standard) which the compiler suite uses.

The CPU registers are illustrated in Figure 10. Note that there is no Program Status Word register shown in this figure. The functions traditionally provided by a PSW register are instead provided in the Status and Cause Registers incorporated within the on-chip System Control Co-Processor (CP0). The instruction set does not use condition codes.

5

Instruction Set Overview

All IDT79R3001 instructions are 32 bits long and there are only three instruction formats (see Figure 11). This approach simplifies decoding, thus minimizing instruction execution time. The IDT79R3001 processor initiates a new instruction on every RUN cycle, and is able to complete an instruction on almost every clock cycle. The only exceptions are the LOAD instructions and BRANCH instructions, which each have a single cycle of latency associated with their execution (that is, the instruction immediately after the branch is always executed regardless of the branch condition; similarly, the data loaded by a LOAD instruction is not available to the subsequent instruction). However, in the majority of cases the compilers (and even the MIPS assembler) are able to reorder instructions to fill these latency cycles with useful instructions which do not require the results of the previous instruction (in the worst case, a NOP instruction is inserted). This effectively eliminates these latency effects and does not require the applications programmer to be aware of the pipeline structure.

The actual instruction set of the CPU was determined after extensive simulations to determine which instructions should be implemented in hardware and which operations are best synthesized in software from other basic operations. This methodology has resulted in the highest performance processor available.

The IDT79R3001 instruction set can be divided into the following groups:

- **Load/Store Instructions** move data between memory and the general registers. These are all "I-Type" instructions. The only addressing mode supported is base register plus signed, immediate 16-bit offset. This effectively allows three addressing modes: register plus offset, register (using zero offset), and immediate (using r0,the zero register).

  The Load instruction has a single cycle of latency, as described above. That is, the instruction immediately after the load instruction cannot rely on the new data; however, the assembler and compilers automatically handle this, reordering code to insure that no conflicts occur. Note that the store operation has no latency in its effect.

  Loads and stores can be performed on byte, half-word, word, or unaligned word data (32-bit data not aligned on a modulo-4 address).
- Computational instructions perform arithmetic, logical, and shift operations on values in registers. They occur in both "R-Type" (both operands and the result are general registers), and "I-Type" (one operand is a 16-bit immediate value) formats.

  Note that computational instructions are three operand instructions: that is, the result register can be different from both source registers. This means that operands need not be overwritten by arithmetic operations. This results in a more efficient use of the register set, and further increases performance.

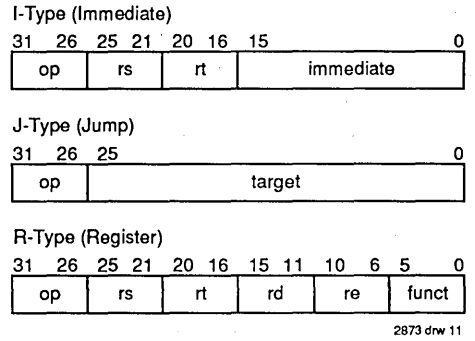


Figure 11. IDT79R3001 Instruction Formats

- Jump and Branch instructions change the flow of control of a program. Jumps are always to a paged absolute address formed by combining a 26-bit target with four bits of the Program Counter ("J-Type" format for subroutine calls), or 32-bit register byte addresses ("R-Type," for Returns and dispatches). Branches have 16-bit offsets relative to the program counter ("I-Type").

  Jump and Link instructions save a return address in Register 31. The IDT79R3001 instruction set features numerous branch conditions. Included is the ability to branch based on a comparison of two registers, or on the comparison of a register to zero. Thus, net performance is increased since the processor does not have to precede the branch instruction with arithmetic operations.
- Co-processor instructions perform operations in the co-processors (such as the IDT79R3010 FPA). Co-processor Loads and Stores are "I- Type;" computational instructions have co-processor dependent formats.
- Co-processor 0 instructions perform operations on the System Control Co-processor (CP0) registers to manipulate the memory management and exception handling facilities of the on-chip co-processor.
- Special instructions perform a variety of tasks, including movement of data between general and special registers, system calls, and breakpoint operations. These are always "R-Type."

IDT79R3001 System Control Co-processor (CP0)

The IDT79R3001 can operate with up to four tightly coupled co-processors, designated CP0-CP3. CP0 is included on-chip as co-processor 0, the System Control co-processor. CP0 is responsible for supporting both the virtual memory system and the exception handling functions of the IDT79R3001.

OP	Description	OP	Description
	Load/Store Instructions		**Multiply/Divide Instructions**
LB	Load Byte	MULT	Multiply
LBU	Load Byte Unsigned	MULTU	Multiply Unsigned
LH	Load Halfword	DIV	Divide
LHU	Load Halfword Unsigned	DIVU	Divide Unsigned
LW	Load Word	MFHI	Move From HI
LWL	Load Word Left	MTHI	Move To HI
LWR	Load Word Right	MFLO	Move From LO
SB	Store Byte	MTLO	Move To LO
SH	Store Halfword		
SW	Store Word		**Jump and Branch Instructions**
SWL	Store Word Left	J	Jump
SWR	Store Word Right	JAL	Jump and Link
		JR	Jump to Register
	Arithmetic Instructions	JALR	Jump and Link Register
	(ALU Immediate)	BEQ	Branch on Equal
ADDI	Add Immediate	BNE	Branch on Not Equal
ADDIU	Add Immediate Unsigned	BLEZ	Branch on Less than or Equal to Zero
SLTI	Set on Less Than Immediate	BGTZ	Branch on Greater Than Zero
SLTIU	Set on Less Than Immediate Unsigned	BLTZ	Branch on Less Than Zero
		BGEZ	Branch on Greater than or Equal to Zero
ANDI	AND Immediate		
ORI	OR Immediate	BLTZAL	Branch on Less Than Zero and Link
XORI	Exclusive OR Immediate	BGEZAL	Branch on Greater than or Equal to Zero and Link
LUI	Load Upper Immediate		
			Special Instructions
	Arithmetic Instructions	SYSCALL	System Call
	(3-operand, register-type)	BREAK	Break
ADD	Add		
ADDU	Add Unsigned		**Coprocessor Instructions**
SUB	Subtract	LWCz	Load Word from Coprocessor
SUBU	Subtract Unsigned	SWCz	Store Word to Coprocessor
SLT	Set on Less Than	MTCz	Move To Coprocessor
		MFCz	Move From Coprocessor
SLTU	Set on Less Than Unsigned	CTCz	Move Control to Coprocessor
AND	AND	CFCz	Move Control From Coprocessor
OR	OR	COPz	Coprocessor Operation
XOR	Exclusive OR	BCzT	Branch on Coprocessor z True
NOR	NOR	BCzF	Branch on Coprocessor z False
	Shift Instructions		**System Control Coprocessor**
SLL	Shift Left Logical		**(CP0) Instructions**
SRL	Shift Right Logical	MTC0	Move To CP0
SRA	Shift Right Arithmetic	MFC0	Move From CP0
SLLV	Shift Left Logical Variable	TLBR	Read indexed TLB entry
SRLV	Shift Right Logical Variable	TLBWI	Write Indexed TLB entry
SRAV	Shift Right Arithmetic Variable	TLBWR	Write Random TLB entry
		TLBP	Probe TLB for matching entry
		RFE	Restore From Exception

2873 tbl 02

Table 2. IDT79R3001 Instruction Summary

5

CP0 Registers

As a co-processor, CP0 has a number of registers which it uses to perform its control functions. These include 64 fully associative Translation Lookaside Buffers (TLBs), used to manage the virtual memory space; registers to manage the TLB set; and the exception handling registers. Figure 12 illustrates the register set of the System Control Co-processor. Table 3 provides a brief explanation of the function of each of these registers. A more detailed explanation of the use of each of these registers is included in the "mips RISC Architecture" manual.

Memory Management System

The IDT79R3001 supports a virtual memory system, so that each task in a given application can be unaware of the addressing needs of other tasks. This is also useful in systems with limited physical memory; the IDT79R3001 provides for the logical expansion of memory by translating addresses composed in a large virtual space into available physical memory addresses.

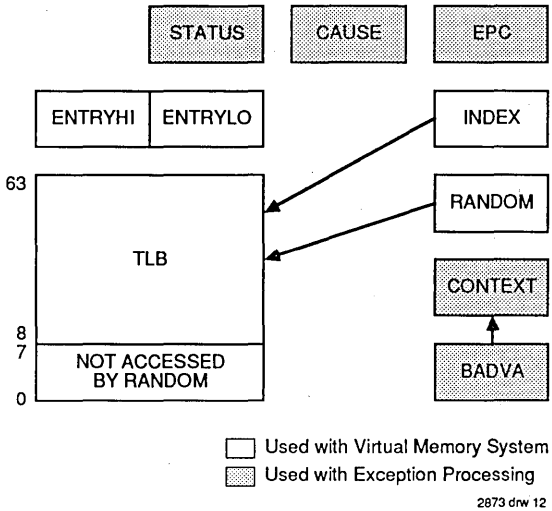


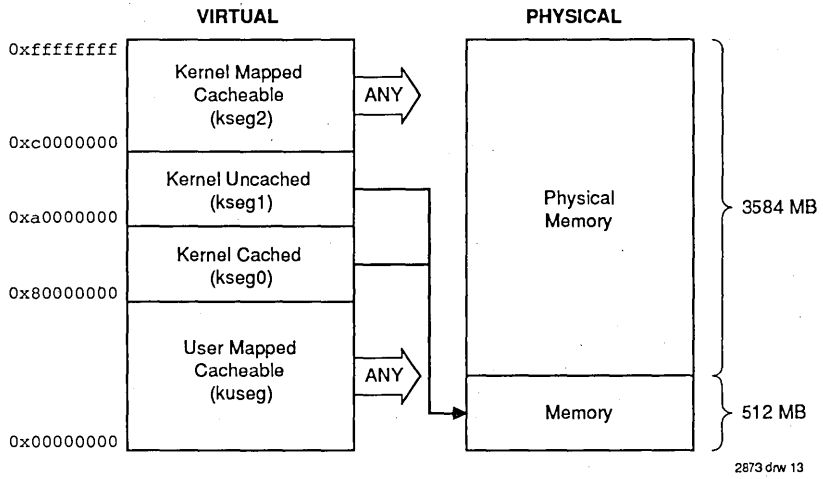
Figure 12. The System Control Co-processor (CP0) Registers

Register	Description
EntryHi	High half of a TLB entry
EntryLo	Low half of a TLB entry
Index	Programmable pointer into TLB array
Random	Pseudo-random pointer into TLB array
Status	Mode, interrupt enables and diagnostic status information
Cause	Indicates nature of last exception
EPC	Exception Program Counter—contains address of instruction which detected the exception
Context	Pointer into kernel's virtual Page Table Entry array
BadVA	Most recent bad virtual address
PrID	Processor revision identification (Read only)

2873 tbl 03

Table 3. CP0 Registers

MMU ADDRESS TRANSLATION



IDT79R3001 Operating Modes

The IDT79R3001 has two operating modes: User Mode and Kernel Mode. The IDT79R3001 normally operates in the User Mode until an exception is detected, forcing it into the Kernel Mode. The processor remains in Kernel Mode until the exceptions are handled and the processor executes an RFE (Return from Exception) instruction, which will restore it to User Mode. Kernel Mode allows software to alter machine state information such as that contained in the CP0 registers; that is, if in User Mode an access is attempted to Co-processor 0 and the Kernel has not enabled the User to access the co-processor, an exception will occur. Similarly, if a User task attempts to use a Kernel virtual address, an exception will occur. Thus, system resources are protected from User tasks.

The manner in which memory addresses are translated (mapped) depends on the operating mode of the IDT79R3001 and on the virtual address desired. Figure 13 illustrates the virtual address mapping performed by the IDT79R3001:

User Mode—in this mode, a single, uniform virtual address space (kuseg) of 2 Gbyte is available to each user task (tasks are further identified by a 6-bit process identifier field in order to form unique virtual addresses). All references to this segment are mapped using the TLB, which utilizes both the virtual address and the Process ID field to perform the virtual-to-physical mapping (note that this allows the cache to be shared by up to 64 User processes at a time without requiring time consuming Cache or TLB flushing).

Kernel Mode—Four separate segments are accessible through this mode:

- **kuseg**—When in the Kernel Mode, references to this segment are treated just like User Mode references, thus streamlining Kernel accesses to User memory.
- **kseg0**—References to this 512 Mbyte segment may use the cache memory, but are not translated by the TLB. Instead, these addresses map directly to the first 512 Mbytes of the physical address space. Note that many dedicated embedded applications will utilize this address space and kseg1only, rather than any of the TLB mapped segments.
- **kseg1**—References to this 512Mbyte segment are not mapped through the TLB. Additionally, this memory is viewed as uncacheable, which means that references through this segment will always use the asynchronous memory interface. As with kseg0, references through this

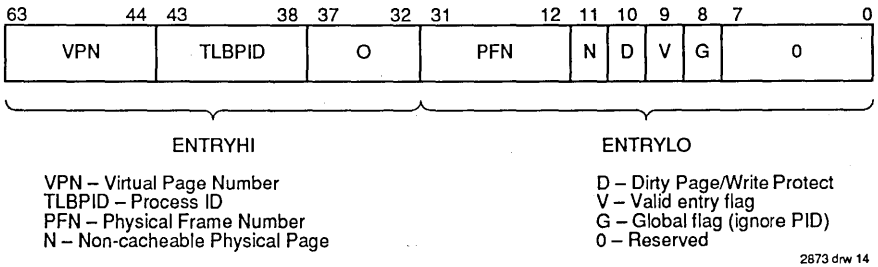


Figure 14. TLB Entry Format

segment are hard-mapped to the first 512 Mbytes of physical memory. When the processor boots, the reset vector is contained in this segment, so that the processor does not require either the cache or the TLB to be valid at RESET time.

- **kseg2**—References to this 1 Gbyte segment are always mapped through the TLB. As with kuseg, the ability of memory pages to be cached is determined by a bit setting in the TLB entry for that page.

The Translation Lookaside Buffer (TLB)

The translation of virtual addresses in either kuseg or kseg2 (mapped segments) is performed by the on-chip Translation Lookaside Buffer array. This array consists of 64 fully-associative (content addressable) memory elements. Each entry maps a 4Kbyte virtual page to a 4Kbyte physical page. Each TLB entry contains other information about the virtual address it maps (such as which User process it maps) and also about the physical address (such as whether it is cacheable or writeable).

Figure 14 illustrates the format of each TLB entry. The translation operation is illustrated in Figure 15. The upper portion of the desired virtual address is compared against the VPN field of each TLB entry. Additionally, the current process ID (contained in the TLBHI register) is matched against the PID field of the TLB entry (if the TLB entry is marked as Global, the PID comparison is ignored). If a match occurs, and the TLB entry is marked as Valid, then the translation is completed by replacing the VPN of the virtual address with the corresponding PFN (Physical Frame Number).

Note that the use of the TLB does not incur an execution penalty, since the execution engine pipeline includes stages to cover for the time required to make the TLB search and translation.

TLB misses occur when no successful match occurs. These events are handled in software. The CP0 registers give the software enough information to obtain the appropriate TLB entry at speeds which exceed those achieved by many CPUs which use hardware TLB replacement (10-12 cycles under UNIX).

When a TLB miss occurs, the address of the instruction which was executing is stored in the EPC register, and the BadVA register contains the address which was being translated. The Context register uses the BadVA value to generate a direct pointer to the kernel Page Table Entry for the desired virtual address. The Random register suggests the TLB entry to be replaced by the new entry. Note that the lower eight TLB entries are not pointed to by Random; the kernel software can thus insure that it is constantly mapped, and deterministic response is guaranteed.

BACKWARD COMPATIBILITY WITH IDT79R2000A AND 79R3000 PROCESSORS

The IDT79R3001 can execute the same binary software (either kernel or user) that is executed by either the IDT79R2000A or IDT79R3000. At the system level, some hardware re-design is necessary to achieve the cost savings inherent in the IDT79R3001 hardware interface.

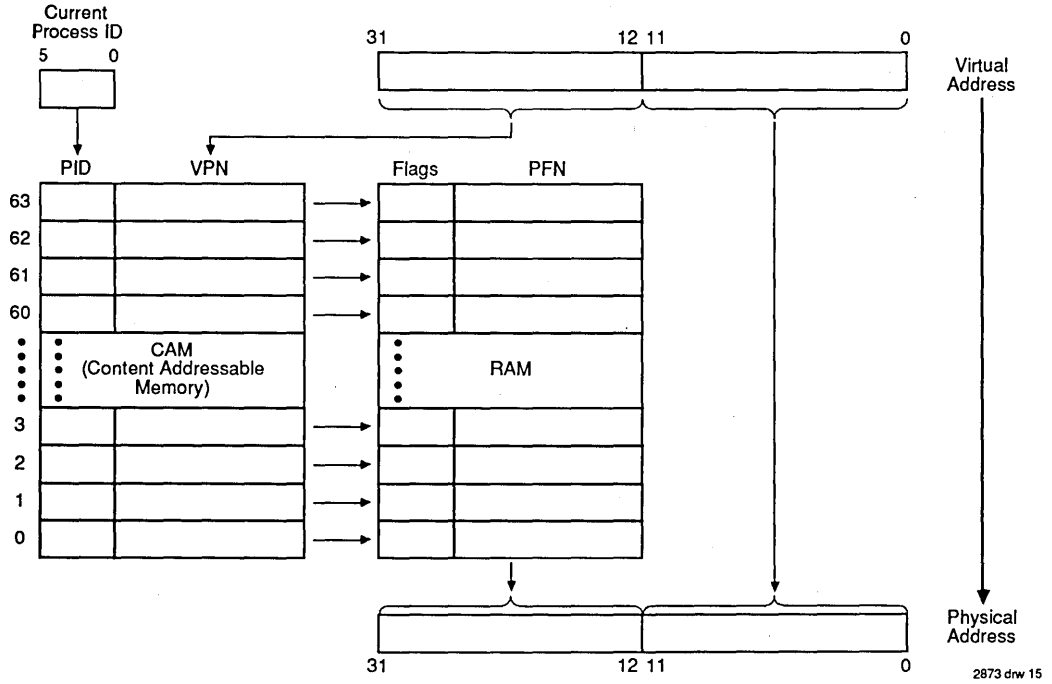


Figure 15. Virtual to Physical TLB Translation

PIN DESCRIPTIONS

Pin Name	I/O	Description
Memory Interface		
Data (0:31)	I/O	A 32-bit bus used for all instruction and data transmission among the processor, synchronous memory space, asynchronous memory space and co-processors.
DataP (0:3)	I/O	A 4-bit bus containing even parity over the data bus. If parity checking is enabled, a parity error will cause the $\overline{\text{PErr}}$ signal to be asserted and a cache-miss to occur. Regardless of whether parity checking is enabled, the processor will always generate parity on writes.
Tag (13:31)	I/O	A 19-bit bus used for transferring cache tags and high-order address bits between the processor, caches and asynchronous memory spaces.
AddrLo (0:23)	O	A 24-bit bus containing low-order byte addresses for both the synchronous (cache) and asynchronous memory spaces.
Synchronous Memory Control		
$\overline{\text{IRd}}$	O	The output enable for the instruction cache. The polarity of this signal is selectable.
$\overline{\text{IWr}}$	O	The write enable for the instruction cache. The polarity of this signal is selectable.
IClk	O	The instruction cache address latch clock. The clock runs continuously.
$\overline{\text{DRd}}$	O	The output enable for the data cache. The polarity of this signal is selectable.
$\overline{\text{DWr}}$	O	The write enable for the data cache. The polarity of this signal is selectable.
DClk	O	The data cache address latch clock. The clock runs continuously.
Valid	I/O	A high on this signal indicates that the Tags just read from the cache are valid. When a cache update occurs, the processor will generate the appropriate Valid bit.
$\overline{\text{PErr}}$	O	If parity checking is enabled, this signal is an active low output of the internal CP0 parity error status bit. It is driven low when a parity error is detected and remains low until software clears the parity error flag in the status register. This pin is physically the same pin as AccTyp2. Its function is selected during device reset.
Asynchronous Memory Interface		
$\overline{\text{XEn}}$	O	The transceiver enable for the read buffer.
AccTyp (0:2)	O	A 3-bit bus used to indicate the size of data being transferred on the asynchronous memory bus, whether or not a data transfer is occurring and the purpose of the transfer. If parity checking is enabled, AccTyp2 becomes the $\overline{\text{PErr}}$ signal.
$\overline{\text{MemWr}}$	O	Signals the occurrence of an asynchronous memory write cycle.
$\overline{\text{MemRd}}$	O	Signals the occurrence of an asynchronous memory read cycle.
$\overline{\text{BusError}}$	I	Signals the occurrence of a bus error during an asynchronous memory transfer cycle.
$\overline{\text{Run}}$	O	Indicates whether the processor is in a RUN or STALL state.
$\overline{\text{Exception}}$	O	Indicates the instruction about to commit processor state should be aborted and other exception related information.
$\overline{\text{SysOut}}$	O	A clock derived from the internal processor clock used to generate the system clock.
RdBusy	I	The asynchronous memory read stall termination signal. In most system designs, RdBusy is normally asserted and is deasserted only to indicate the successful completion of the memory read. RdBusy is sampled by the processor only during memory read stalls.
$\overline{\text{WrBusy}}$	I	The asynchronous memory write stall initiation/termination signal. $\overline{\text{WrBusy}}$ is only sampled during write operation.
Co-Processor Interface		
$\overline{\text{CpSync}}$	O	A clock which is identical to $\overline{\text{SysOut}}$ and used by co-processors for timing synchronization with the CPU.
CPBusy	I	The co-processor busy stall initiation/termination signal.
CpCond (0:3)	I	A 4-bit bus used to transfer conditional branch status from the co-processors to the CPU. CpCond(0) is used to control whether or not a cache burst refill occurs; the other signals are used as input port pins for co-processor branch instructions.
Processor Control Signals		
DMAStall	I	DMA Stall. Signals to the processor that it should stall accesses to the synchronous memories and tri-state the synchronous memory interface.
$\overline{\text{Int}}$ (0:5)	I	A 6-bit bus used to signal maskable interrupts to the CPU. A reset time, mode values are sampled from this bus to initialize the processor. During normal operation, these signals are not latched by the processor and must remain asserted until the processor acknowledges the interrupt (through software) to the interrupt source.
Clk2xSys	I	The master double frequency input clock, used to generate $\overline{\text{SysOut}}$.
Clk2xSmp/Rd	I	A double frequency clock input used to determine the sample point for data coming into the CPU and co-processors and used to determine the enable time of the synchronous memory RAMs.
Clk2xPhi	I	A double frequency clock input used to determine the position of the two internal phases.
$\overline{\text{Reset}}$	I	Initialization input used to force execution starting from the reset memory address. $\overline{\text{Reset}}$ should be asserted asynchronously but must be negated synchronously with the leading edge of $\overline{\text{SysOut}}$.

ABSOLUTE MAXIMUM RATINGS[1, 3]

Symbol	Rating	Commercial	Military	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70[4] (Ambient) 0 to +90[5] (Case)	–55 to +125 (Case)	°C
TBIAS	Case Temperature Under Bias	–55 to +125[4] 0 to +90[5]	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
IIN	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

2873 tbl 05

NOTE:

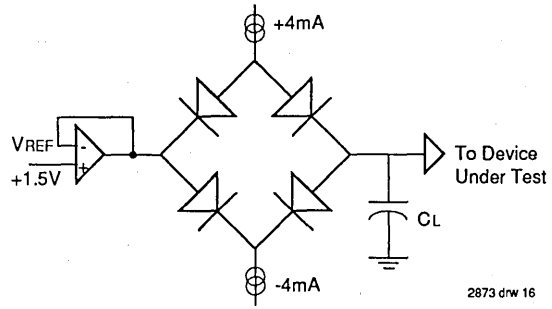
1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. VIN minimum = -3.0V for pulse width less than 15ns. VIN should not exceed VCC +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.
4. 16-33 MHz only.
5. 37-40 MHz only.

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	VCC
Military 16-33 MHZ	-55°C to +125°C (Case)	0V	5.0 ±10%
Commercial 16-33 MHz	0°C to +70°C (Ambient)	0V	5.0 ±5%
Commercial 37-40 MHz	0°C to +90°C (Case)	0V	5.0 ±5%

2873 tbl 07

OUTPUT LOADING FOR AC TESTING



2873 drw 16

AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V

2873 tbl 06

Signal	CL
IRd, IWr, DRd, DWr	50pf
All Others	25pf

DC ELECTRICAL CHARACTERISTICS
COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHT}	Output HIGH Voltage(4,7)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OHC}	Output HIGH Voltage(8)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OLT}	Output LOW Voltage(4,7)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
I_{RESET}	Input HIGH Current(6)		10	100	10	100	10	100	10	100	µA
C_{IN}	Input Capacitance(7)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = Max.	—	575	—	650	—	750	—	800	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	-100	100	-100	100	-100	100	-100	100	µA

2873 tbl 08

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp/Rd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag buses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are supplies as additional information to help the system designer understand the relationship between current drive and output voltage on these pins..
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. The IDT79R3001 contains an internal pull-up/current source on the TAG pins to facilitate initialization. This current source is disconnected when $\overline{\text{Reset}}$ is inactive.
7. Guaranteed by design.
8. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

5

DC ELECTRICAL CHARACTERISTICS
COMMERCIAL TEMPERATURE RANGE (T_C = 0°C to +90°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	37.0MHz		40.0MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	-0.4	—	-0.4	V
I_{RESET}	Input HIGH Current(6)		10	100	10	100	µA
C_{IN}	Input Capacitance(6)		—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	825	—	850	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	µA

2873 tbl 09

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag buses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

DC ELECTRICAL CHARACTERISTICS
MILITARY TEMPERATURE RANGE (T_C = -55°C to +125°C, V_{CC} = +5.0V ±10%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHT}	Output HIGH Voltage(4,7)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OHC}	Output HIGH Voltage(8)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OLT}	Output LOW Voltage(4,7)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
I_{RESET}	Input HIGH Current(6)		10	100	10	100	10	100	10	100	µA
C_{IN}	Input Capacitance(7)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = Max.	—	500	—	600	—	650	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	-100	100	-100	100	-100	100	-100	100	µA

NOTES:

2873 tbl 08

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp/Rd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag buses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are supplies as additional information to help the system designer understand the relationship between current drive and output voltage on these pins..
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. The IDT79R3001 contains an internal pull-up/current source on the TAG pins to facilitate initialization. This current source is disconnected when $\overline{\text{Reset}}$ is inactive.
7. Guaranteed by design.
8. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

5

AC ELECTRICAL CHARACTERISTICS[1,4]

COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Clock											
TckHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2xSmp/Rd[5]		0	Tcyc/4	0	Tcyc/4	0	Tcyc/4	0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi[5]		9	Tcyc/4	7	Tcyc/4	5	Tcyc/4	3.5	Tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	2	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type2	Load= 25pF	17	—	14	—	12	—	—	8.5	ns
TMWr	Memory Write	Load= 25pF	1	27	1	23	1	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	1	18	1	13.5	ns
TMRdT	Memory Read Terminate	Load= 25pF	1	2	1	23	1	5	1	13.5	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSEc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
TDMADis	DMA Drive On	Load= 25pF	3	15	3	15	3	15	3	15	ns
TDMAEn	DMA Drive Off	Load= 25pF	—	10	—	10	—	10	—	10	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		140	—	140	—	140	—	140	—	µs
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	1	0.5	1	0.5	1	0.5	1	ns/25pF

2873 tbl 11

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all three 2xClocks: Clk2xSys, Clk2xSmp/Rd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters are illustrated in detail in the "IDT79R3001 Hardware Interface Guide".
5. Tcyc is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of $\overline{\text{Run}}$, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Transition time <2.5ns for 33MHz; <5ns for lower speeds.

AC ELECTRICAL CHARACTERISTICS[1,4]

COMMERCIAL TEMPERATURE RANGE (Tc = 0°C to +90°C, Vcc = +5.0V ±5%)

Symbol	Parameter	Test Conditions	37.0MHz		40.0MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
TckHigh	Input Clock High[2]	Transition < 2.5ns	5.5	—	5	—	ns
TckLow	Input Clock Low[2]	Transition < 2.5ns	5.5	—	5	—	ns
TckP	Input Clock Period[2]		13.5	500	12.5	500	ns
	Clk2xSys to Clk2xSmp/Rd[5]		0	Tcyc/4	0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi[5]		3.5	Tcyc/4	3	Tcyc/4	ns
Run Operation							
TDEn	Data Enable[3]		—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	1.5	ns
TWrDly	Write Delay	Load= 25pF	—	2	—	2	ns
TDS	Data Set-up		4.5	—	4	—	ns
TDH	Data Hold		-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		6	—	6	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	3.5	—	3	ns
TAT2	Access Type2	Load= 25pF	—	8.5	—	7.5	ns
TMWr	Memory Write	Load= 25pF	—	9.5	—	9	ns
TExc	Exception	Load= 25pF	—	3.5	—	3	ns
Stall Operation							
TSAVal	Address Valid	Load= 25pF	—	15	—	12.5	ns
TSAcTy	Access Type	Load= 25pF	—	13.5	—	9	ns
TMRdi	Memory Read Initiate	Load= 25pF	—	13.5	—	9	ns
TMRdT	Memory Read Terminate	Load= 25pF	—	10	—	9	ns
TStl	Run Terminate	Load= 25pF	2	6.5	2	6	ns
TRun	Run Initiate	Load= 25pF	—	3	—	3	ns
TSMWr	Memory Write	Load= 25pF	2	9.5	2	9	ns
TSExc	Exception Valid	Load= 25pF	—	6.5	—	6	ns
TDMADis	DMA Drive On	Load= 25pF	3	15	3	15	ns
TDMAEn	DMA Drive Off	Load= 25pF	—	10	—	10	ns
Reset Initialization							
TRST	Reset Pulse Width		—	—	—	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		—	—	—	—	µs
Capacitive Load Deration							
CLD	Load Derate[6]		—	—	—	—	ns/25pF

NOTES:

2873 tbl 12

1. All timings are referenced to 1.5V.
2. The clock parameters apply to all three 2xClocks: Clk2xSys, Clk2xSmp/Rd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters are illustrated in detail in the "IDT79R3001 Hardware Interface Guide".
5. Tcyc is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of $\overline{\text{Run}}$, no two signals on a given device will derate for a given load by a difference greater than 15%.

5

AC ELECTRICAL CHARACTERISTICS[1,4]

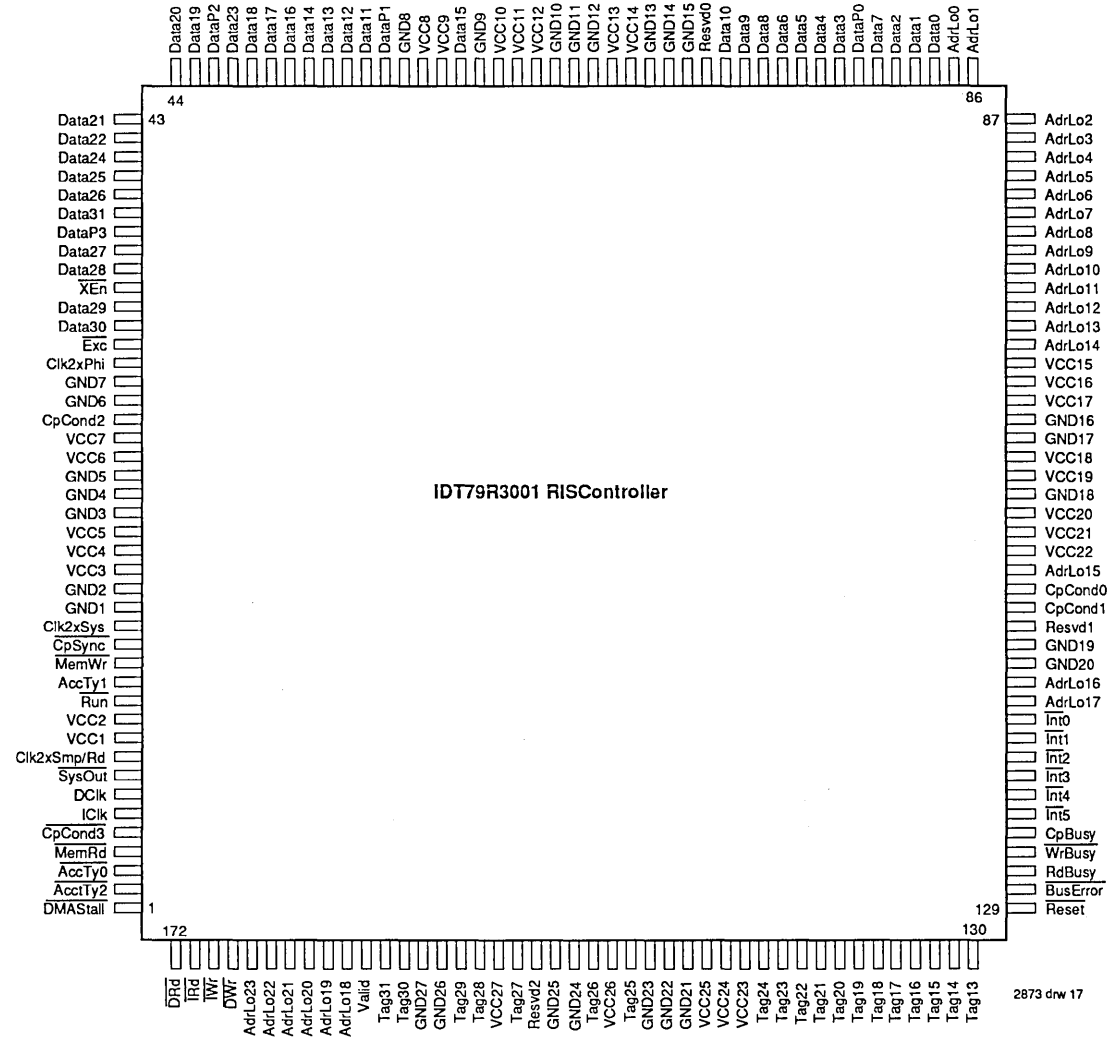
MILITARY TEMPERATURE RANGE (Tc = -55°C to +125°C, Vcc = +5.0V ±10%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Clock											
TckHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2xSmp/Rd[5]		0	Tcyc/4	0	Tcyc/4	0	Tcyc/4	0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi[5]		9	Tcyc/4	7	Tcyc/4	5	Tcyc/4	3.5	Tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	2	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type2	Load= 25pF	17	—	14	—	12	—	—	8.5	ns
TMWr	Memory Write	Load= 25pF	1	27	1	23	1	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	1	18	1	13.5	ns
TMRdT	Memory Read Terminate	Load= 25pF	1	2	1	23	1	5	1	13.5	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSEc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
TDMADis	DMA Drive On	Load= 25pF	3	15	3	15	3	15	3	15	ns
TDMAEn	DMA Drive Off	Load= 25pF	—	10	—	10	—	10	—	10	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		140	—	140	—	140	—	140	—	µs
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	1	0.5	1	0.5	1	0.5	1	ns/25pF

2873 tbl 13

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all three 2xClocks: Clk2xSys, Clk2xSmp/Rd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters are illustrated in detail in the "IDT79R3001 Hardware Interface Guide".
5. Tcyc is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of $\overline{\text{Run}}$, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Transition time <2.5ns for 33MHz; <5ns for lower speeds.

PIN CONFIGURATIONS
172-Pin Ceramic Flatpack (Cavity Side View)



2873 drw 17

NOTE:
1. AccTyp2 is redefined to be $\overline{\text{Parity Error}}$ if the parity enable option is selected at device initialization.

PIN CONFIGURATIONS (Continued)
144-Pin PGA (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	VCC14	AdrLo 6	AdrLo 10	AdrLo 11	VCC12	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	$\overline{\text{Int(2)}}$	$\overline{\text{Int(5)}}$	$\overline{\text{Wr}}$ Busy	$\overline{\text{Reset}}$	VCC10
B	AdrLo 3	$\overline{\text{Mem}}$ Wr	AdrLo 7	AdrLo 9	AdrLo 12	$\overline{\text{Cp}}$ Sync	AdrLo 13	CpCond 1	$\overline{\text{Int(1)}}$	$\overline{\text{Int(3)}}$	Cp Busy	$\overline{\text{Bus}}$ Error	$\overline{\text{Run}}$	Tag13	Tag16
C	AdrLo 0	AdrLo 4	VCC13	AdrLo 5	AdrLo 8	GND13	GND12	VCC11	$\overline{\text{Int(0)}}$	$\overline{\text{Int(4)}}$	Rd Busy	GND	Tag14	Tag17	Tag20
D	Data 1	AdrLo 2	GND0										Tag15	Tag19	Tag21
E	DataP 0	Data 0	AdrLo 1										Tag18	Tag22	VCC9
F	VCC0	Data 7	Data 2										GND10	Tag23	Tag25
G	Data 4	Data 3	GND1										GND9	Tag24	Tag26
H	Data 6	Data 5	Data 8					**IDT79R3001 RISController**					VCC8	Tag28	Tag27
J	Data 10	DataP 1	Data 9										Tag31	Valid	Tag29
K	Data 15	Data 11	GND2										GND8	AdrLo 19	Tag30
L	VCC1	Data 12	Data 17										AdrLo 22	AdrLo 20	AdrLo 18
M	Data 13	Data 16	DataP 2										GND7	AdrLo 23	VCC7
N	Data 14	Data 18	Data 19	GND3	Data 24	DataP 3	VCC3	VCC4	GND5	GND6	$\overline{\text{Mem}}$ Rd	DMA Stall	$\overline{\text{DRd}}$	$\overline{\text{IWr}}$	AdrLo 21
P	Data 23	Data 20	AccTy1	Data 22	Data 26	Data 27	$\overline{\text{XEn}}$	Data 30	Clk2x Sys	Clk2x Smp/Rd	$\overline{\text{DClk}}$	Cp Cond3	AccTy0	$\overline{\text{IRd}}$	$\overline{\text{DWr}}$
Q	VCC2	Data 21	Data 25	Data 31	Data 28	GND4	Data 29	$\overline{\text{Excep-}}$ tion	Clk2x Phi	Cp Cond2	$\overline{\text{SysOut}}$	VCC5	IClk	AccTy2	VCC6

2873 drw 18

NOTE:
1. AccTyp2 is redefined to be $\overline{\text{Parity Error}}$ if the parity enable option is selected at device initialization.

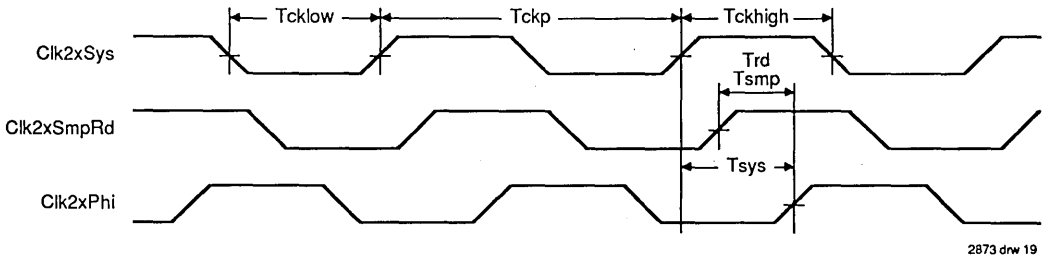
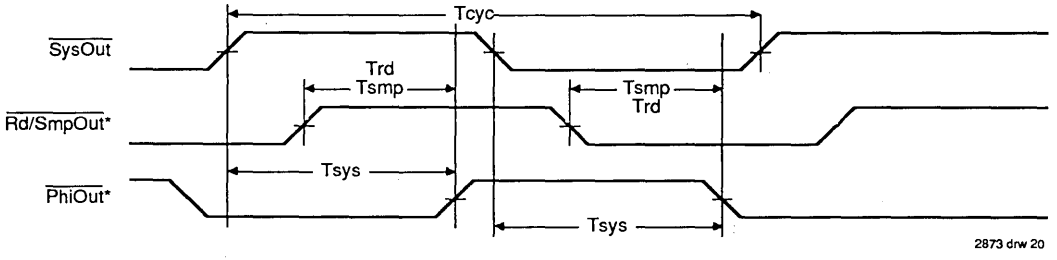


Figure 16. Input Clock Timing



* These signals are not actually output from the processor. They are drawn to provide a reference for other timing diagrams.

Figure 17. Processor Reference Clock Timing

5

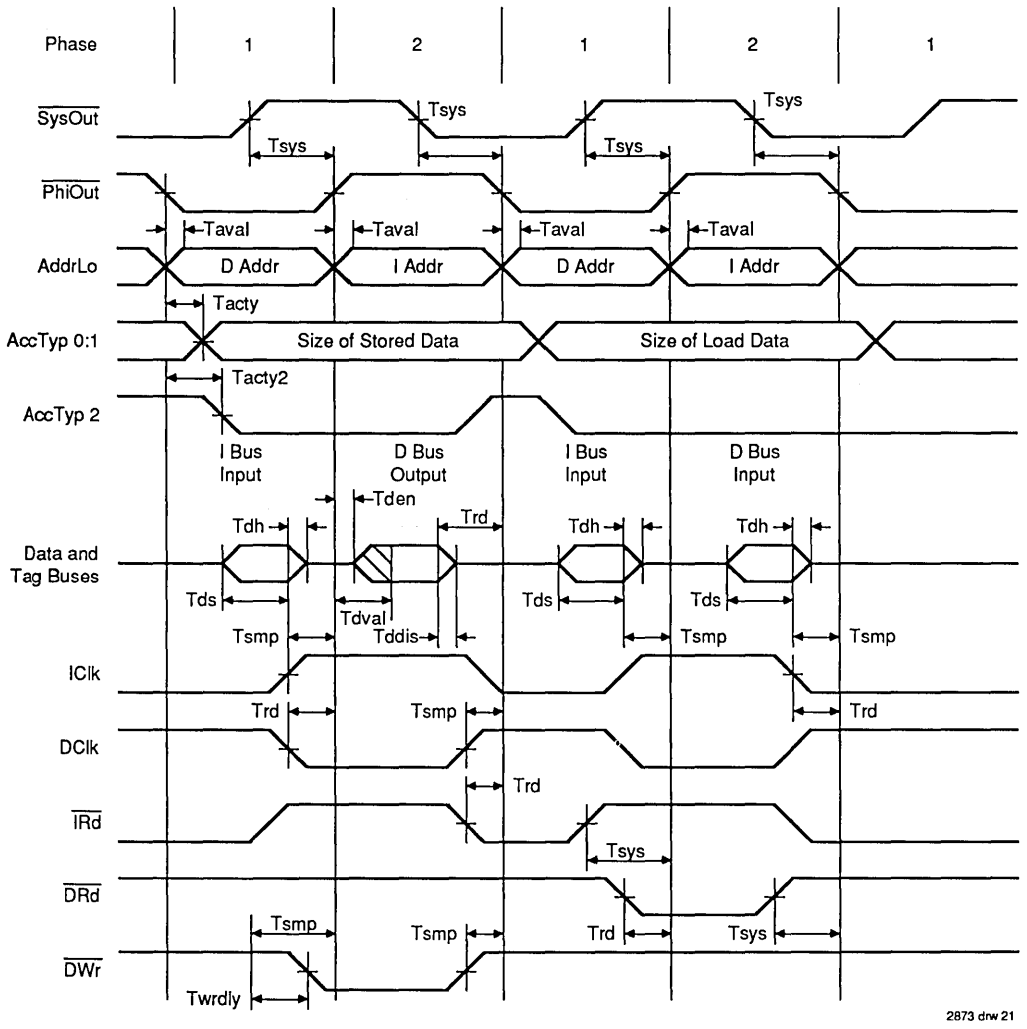


Figure 18. Synchronous Memory (Cache) Timing

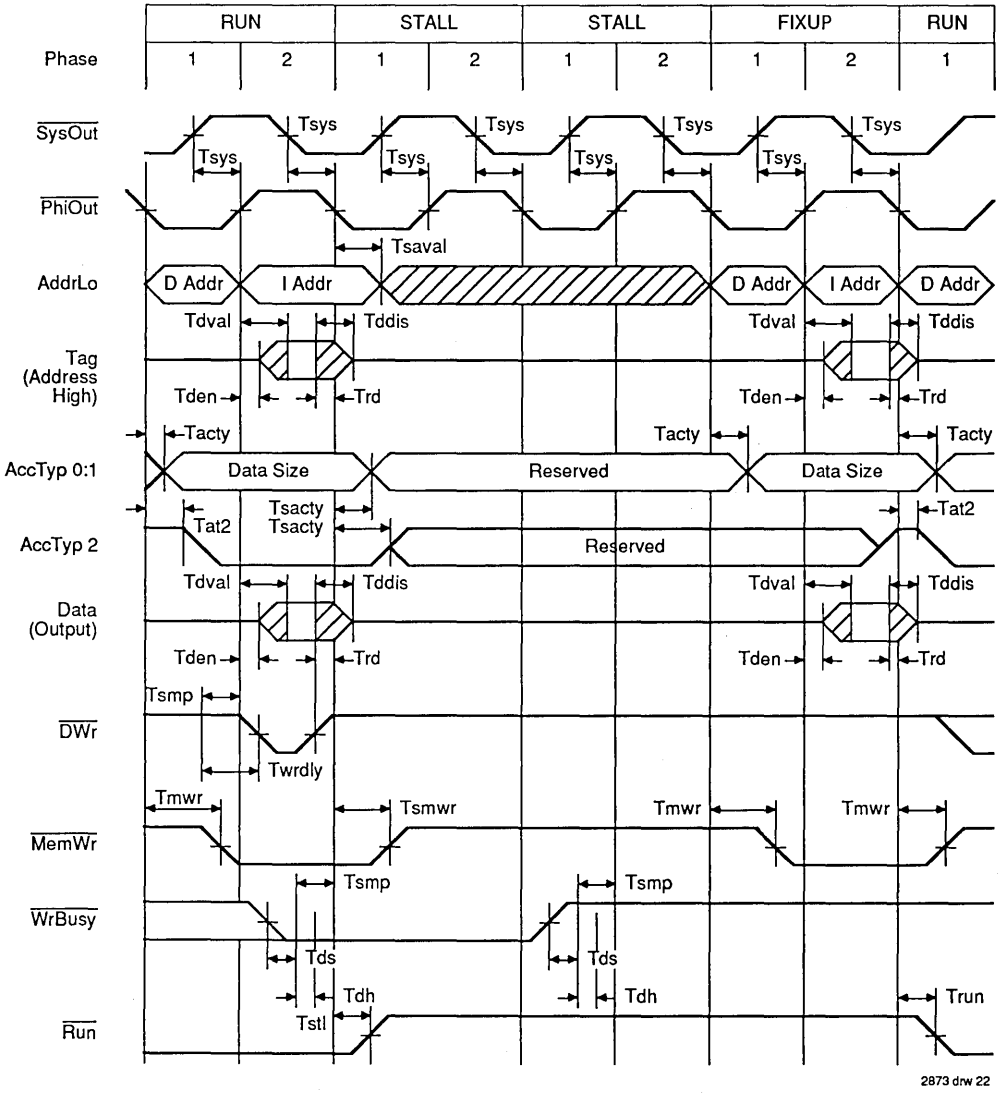


Figure 19. Memory Write Timing

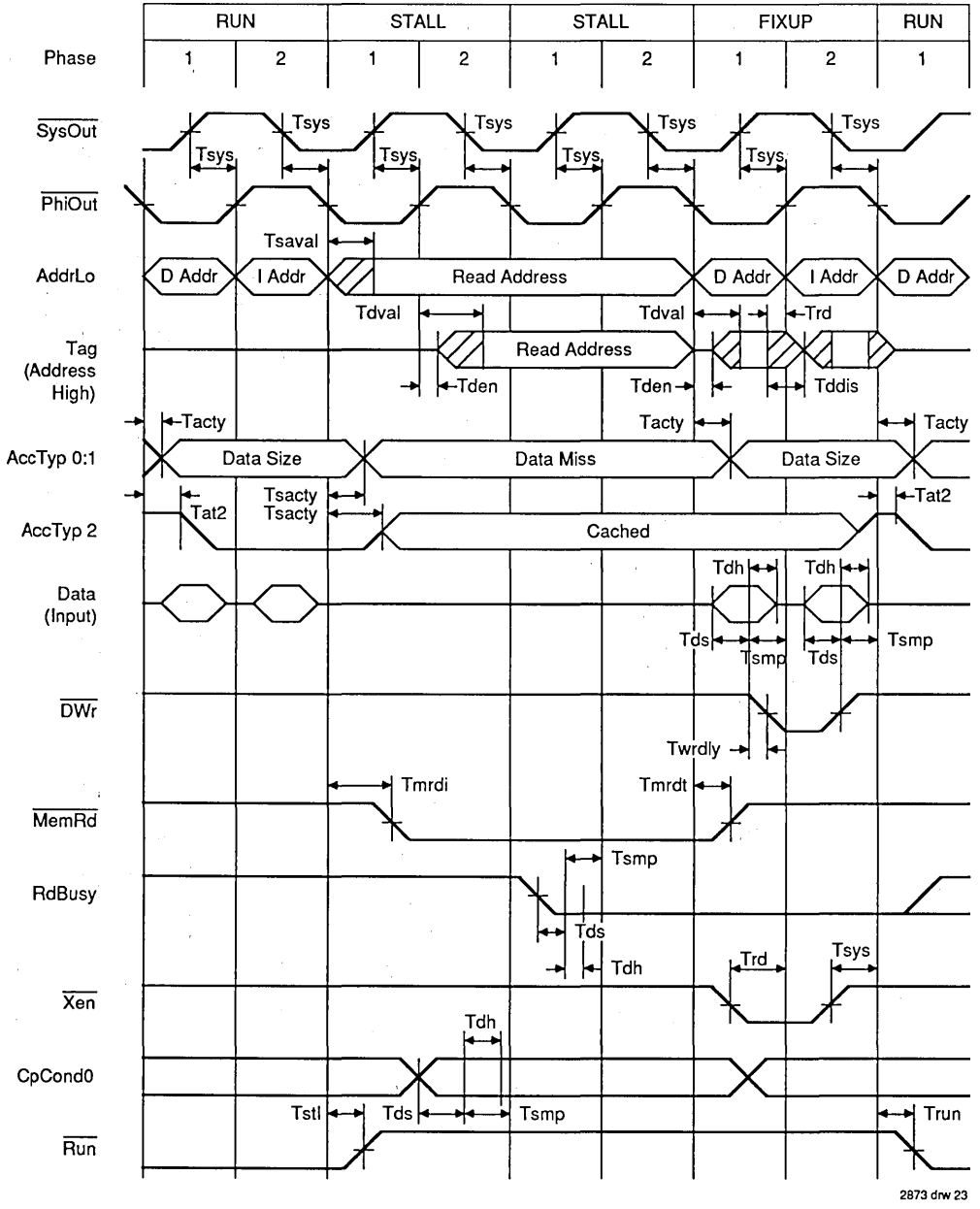


Figure 20. Memory Read Timing

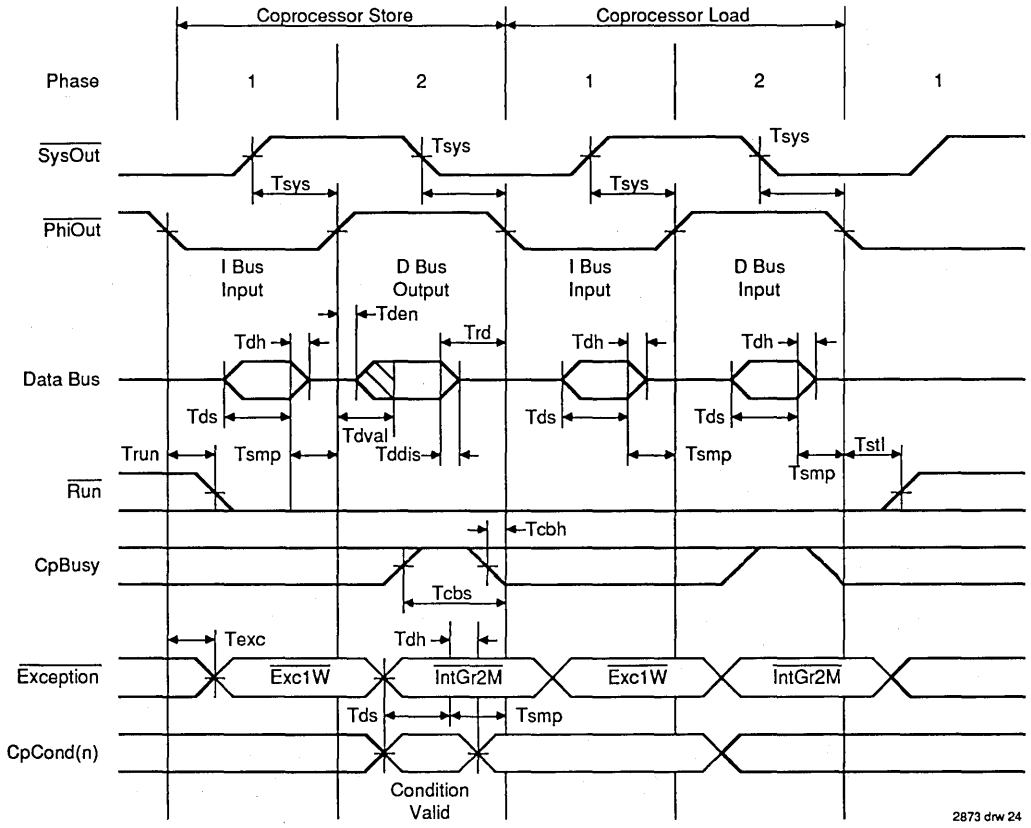


Figure 21. Co-Processor Load/Store Timing

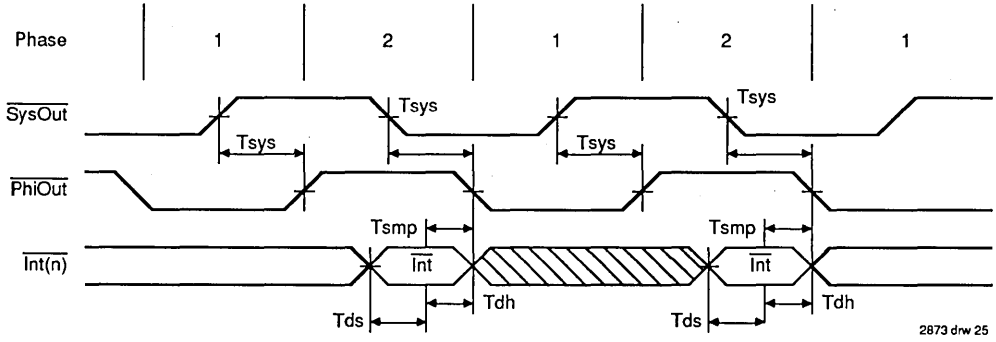


Figure 22. Interrupt Timing

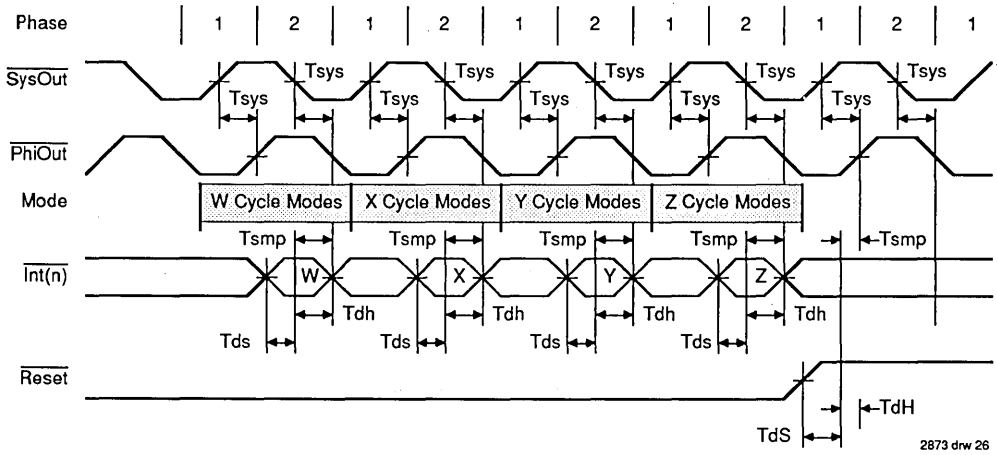


Figure 23. Mode Vector Initialization

NOTES:
1. Reset must be negated synchronously; however, it can be asserted asynchronously. Designs must not rely on the proper functioning of $\overline{\text{SysOut}}$ prior to the assertion of $\overline{\text{Reset}}$.
2. If $\overline{\text{Phase-Lock On}}$ or is asserted as mode select options, they should be asserted throughout the $\overline{\text{Reset}}$ period, to insure that the slowest coprocessor in the system has sufficient time to lock to the CPU clocks.
3. $\overline{\text{Reset}}$ is actually sampled in both Phase 1 and Phase 2. To insure proper initialization, it must be negated relative to the end of Phase 1.

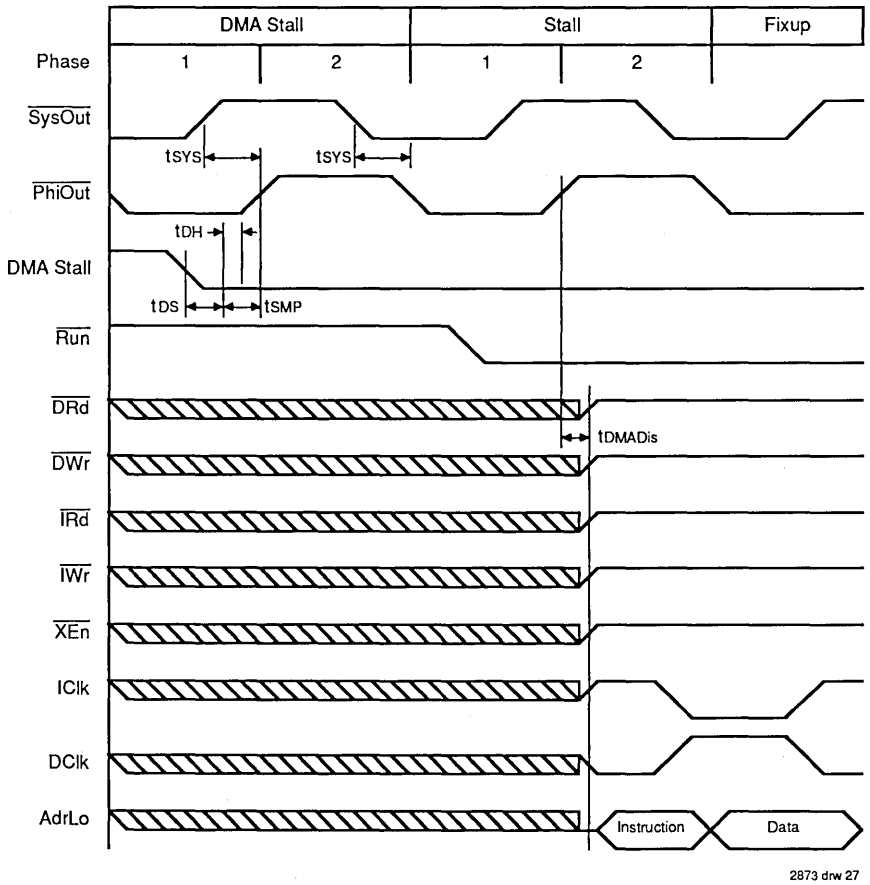


Figure 24. Entering DMA Stall

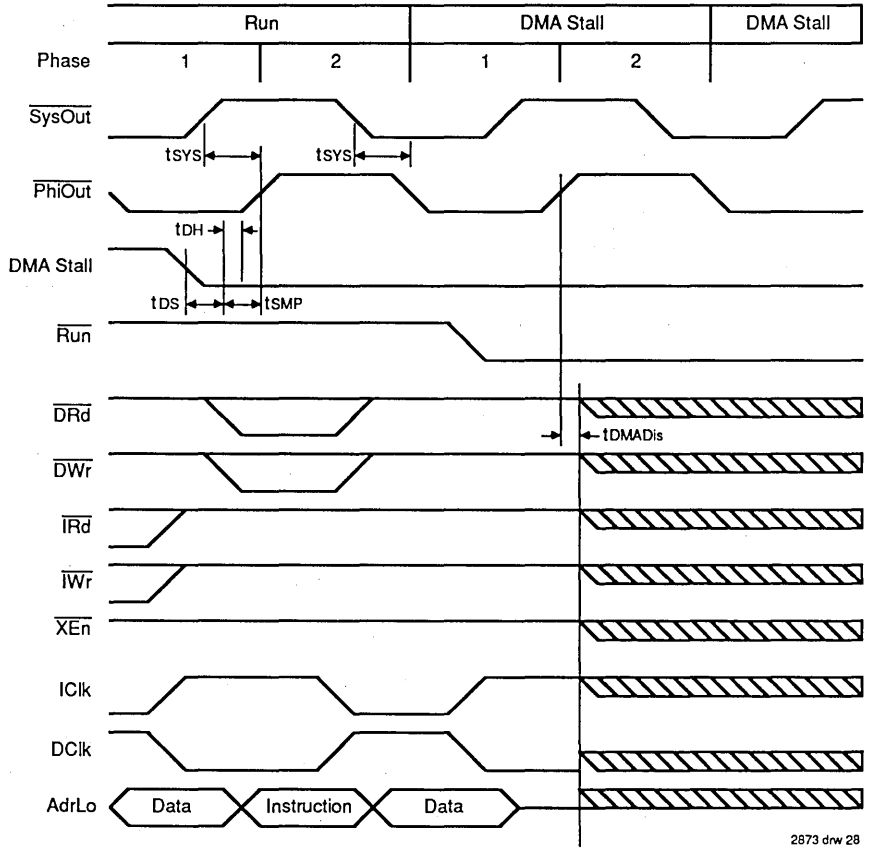
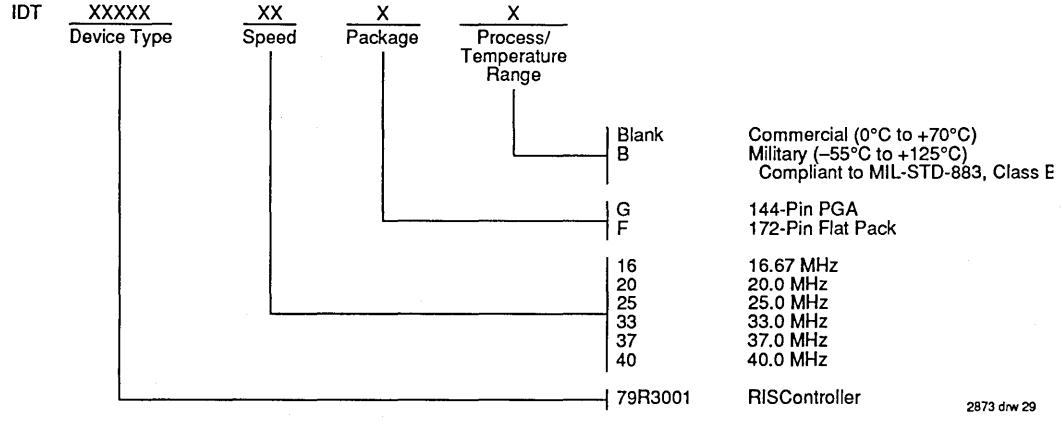


Figure 25. Completing DMA Stall

ORDERING INFORMATION



Integrated Device Technology, Inc.

RISC FLOATING POINT ACCELERATOR (FPA)

IDT79R3010A
IDT79R3010AE

FEATURES:

- Hardware Support of Single- and Double-Precision Operations:
  - — Floating-Point Add
  - — Floating-Point Subtract
  - — Floating-Point Multiply
  - — Floating-Point Divide
  - — Floating-Point Comparisons
  - — Floating-Point Conversions
- Sustained performance:
  - — 11 MFLOPS single precision LINPACK
  - — 7.3 MFLOPS double precision LINPACK
- 16.7MHz through 40 MHz operation
- Direct, high-speed interface with IDT79R3000A and IDT79R3001 Processor
- Supports Full Conformance With IEEE 754-1985 Floating-Point Specification
- Full 64-bit operation using sixteen 64-bit data registers
- High-speed CEMOS™ technology
- Military product compliant to MIL-STD-883, Class B
- 32-bit status/control register providing access to all IEEE-Standard exception handling
- Load/store architecture allows data movement directly between FPA and memory or between CPU and FPA
- Overlapped operation of independent floating point ALUs
- Fully pin-compatible with IDT79R3010/IDT79R3010L

DESCRIPTION:

The IDT79R3010A Floating-Point Accelerator (FPA) operates in conjunction with the IDT79R3000A Processor and extends the IDT79R3000A's instruction set to perform arithmetic operations on values in floating-point representations. The IDT79R3010A FPA, with associated system software, fully conforms to the requirements of ANSI/IEEE Standard 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic." In addition, the architecture fully supports the standard's recomendations.

This data sheet provides an overview of the features and architecture of the 79R3010A FPA. A more detailed description of the operation of the device is incorporated in the "R3000A Family Hardware User's Manual," and a more detailed architectural overview is provided in the "mips RISC Architecture" book, both available from IDT.

5

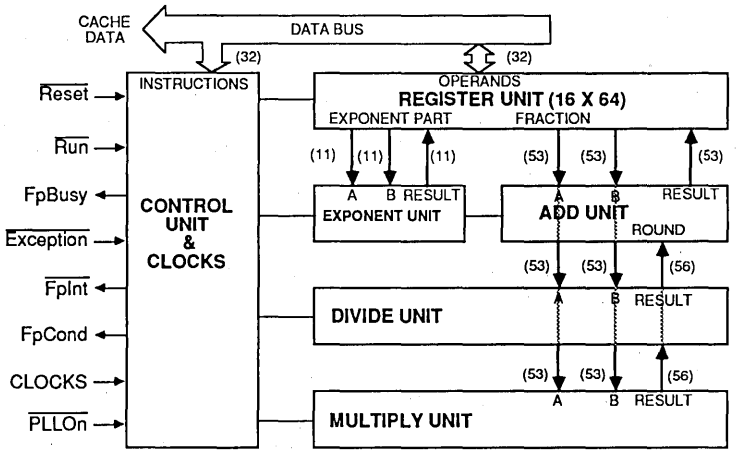


Figure 1. IDT79R3010A Functional Block Diagram

CEMOS is a trademark of Integrated Device Technology, Inc.

IDT79R3010A FPA REGISTERS

The IDT79R3010A FPA provides 32 general purpose 32-bit registers, a Control/Status register, and a Revision Identification register. The tightly-coupled coprocessor interface causes the register resources of the FPA to appear to the systems programmers as an extension of the CPU internal registers. The FPA registers are shown in Figure 2.

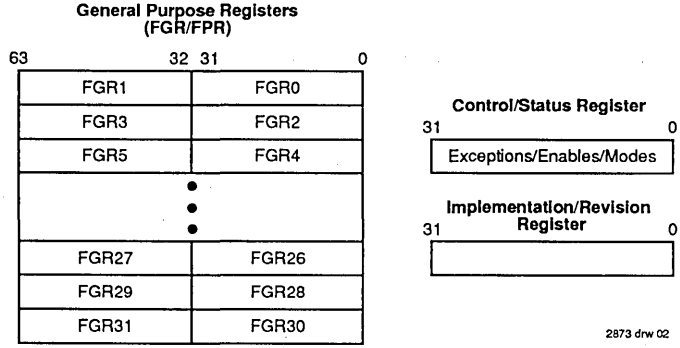


Figure 2. IDT79R3010A FPA Registers

Floating-point coprocessor operations reference three types of registers:

- Floating-Point Control Registers (FCR)
- Floating-Point General Registers (FGR)
- Floating-Point Registers (FPR)

Floating-Point General Registers (FGR)

There are 32 Floating-Point General Registers (FGR) on the FPA. They represent directly-addressable 32-bit registers, and can be accessed by Load, Store, or Move Operations.

Floating-Point Registers (FPR)

The 32 FGRs described in the preceding paragraph are also used to form sixteen 64-bit Floating-Point Registers (FPR). Pairs of general registers (FGRs), for example FGR0 and FGR1 (refer to Figure 2) are physically combined to form a single 64-bit FPR. The FPRs hold a value in either single- or double-precision floating-point format. Double-precision format FPRs are formed from two adjacent FGRs.

Floating-Point Control Registers (FCR)

There are 2 Floating-Point Control Registers (FCR) on the FPA. They can be accessed only by Move operations and include the following:

- Control/Status register, used to control and monitor exceptions, operating modes, and rounding modes;
- Revision register, containing revision information about the FPA.

COPROCESSOR OPERATION

The FPA continually monitors the IDT79R3000A processor instruction stream. If an instruction does not apply to the coprocessor, it is ignored; if an instruction does apply to the coprocessor, the FPA executes that instruction and transfers necessary result and exception data synchronously to the IDT79R3000A main processor.

The FPA performs three types of operations:

- Loads and Stores;
- Moves;
- Two- and three-register floating-point operations.

Load, Store, and Move Operations

Load, Store, and Move operations move data between memory or the IDT79R3000A Processor registers and the IDT79R3010A FPA registers. These operations perform no format conversions and cause no floating-point exceptions. Load, Store, and Move operations reference a single 32-bit word of either the Floating-Point General Registers (FGR) or the Floating-Point Control Registers (FCR).

Floating-Point Operations

The FPA supports the folowing single- and double-precision format floating-point operations:

- Add
- Subtract
- Multiply
- Divide
- Absolute Value
- Move
- Negate
- Compare

In addition, the FPA supports conversions between single- and double-precision floating-point formats and fixed-point formats.

The FPA incorporates separate Add/Subtract, Multiply, and Divide units, each capable of independent and concurrent operation. Thus, to achieve very high performance, floating point divides can be overlapped with floating point multiplies and floating point additions. These floating point operations occur independently of the actions of the CPU, allowing further overlap of integer and floating point operations. Figure 3 illustrates an example of the types of overlap permissible.

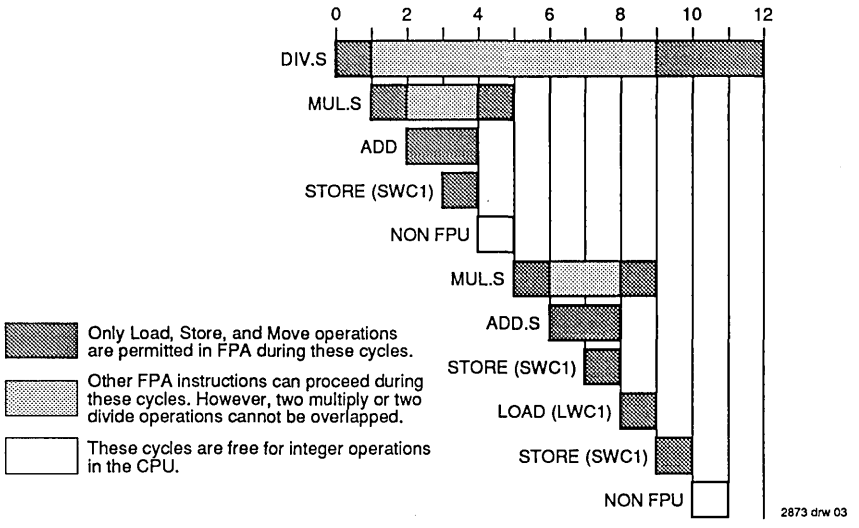


Figure 3. Examples of Overlapping Floating Point Operation

Exceptions

The IDT79R3010A FPA supports all five IEEE standard exceptions:

- Invalid Operation
- Inexact Operation
- Division by Zero
- Overflow
- Underflow

The FPA also supports the optional, Unimplemented Operation exception that allows unimplemented instructions to trap to software emulation routines.

The FPA provides precise exception capability to the CPU; that is, the execution of a floating point operation which generates an exception causes that exception to occur at the CPU instruction which caused the operation. This precise exception capability is a requirement in applications and languages which provide a mechanism for local software exception handlers within software modules.

INSTRUCTION SET OVERVIEW

All IDT79R3010A instructions are 32 bits long and they can be divided into the folowing groups:

- **Load/Store and Move** instructions move data between memory, the main processor and the FPA general registers.
- **Computational** instructions perform arithmetic operations on floating point values in the FPA registers.
- **Conversion** instructions perform conversion operations between the various data formats.
- **Compare** instructions perform comparisons of the contents of registers and set a condition bit based on the results. The result of the compare operation is output on the FpCond output of the FPA, which is typically used as CpCond1 on the CPU for use in coprocessor branch operations.

Table 1 lists the instruction set of the IDT79R3010A FPA.

OP	Description	OP	Description
	Load/Store/Move Instructions		**Computational Instructions**
LWC1	Load Word to FPA	**ADD.fmt**	Floating-point Add
SWC1	Store Word from FPA	**SUB.fmt**	Floating-point Subtract
MTC1	Move Word to FPA	**MUL. fmt**	Floating-point Multiply
MFC1	Move Word from FPA	**DIV.fmt**	Floating-point Divide
CTC1	Move Control word to FPA	**ABS.fmt**	Floating-point Absolute value
CFC1	Move Control word from FPA	**MOV.fmt**	Floating-point Move
		NEG.fmt	Floating-point Negate
	Conversion Instructions		**Compare Instructions**
CVT.S.fmt	Floating-point Convert to Single FP	**C.cond.fmt**	Floating-point Compare
CVT.D.fmt	Floating-point Convert to Double FP		
CVT.W.fmt	Floating-point Convert to fixed-point		

2873 tbl 01

Table 1. IDT79R3010A Instruction Summary

ID79R3010 PIPELINE ARCHITECTURE

The IDT79R3010A FPA provides an instruction pipeline that parallels that of the IDT79R3000A processor. The FPA, however, has a 6-stage pipeline instead of the 5-stage pipeline of the IDT79R3000: the additional FPA pipe stage is used to provide efficient coordination of exception responses between the FPA and main processor.

The execution of a single IDT79R3010A instruction consists of six primary steps:

1) **IF**—Instruction Fetch. The main processor calculates the insruction address required to read an instruction from the I-Cache. No action is required of the FPA during this pipe stage since the main processor is responsible for address generation.
2) **RD**—The instruction is present on the data bus during phase 1 of this pipe stage and the FPA decodes the instruction on the bus to determine if it is an instruction for the FPA.
3) **ALU**—If the instruction is an FPA instruction, instruction execution commences during this pipe stage.
4) **MEM**—If this is a coprocessor load or store instruction, the FPA presents or captures the data during phase 2 of this pipe stage.
5) **WB**—The FPA uses this pipe stage solely to deal with exceptions.
6) **FWB**—The FPA uses this stage to write back ALU results to its register file. This stage is the equivalent of the WB stage in the IDT79R3000A main processor.

Each of these steps requires approximately one FPA cycle as shown in Figure 3 (parts of some operations spill over into another cycle while other operations require only 1/2 cycle).

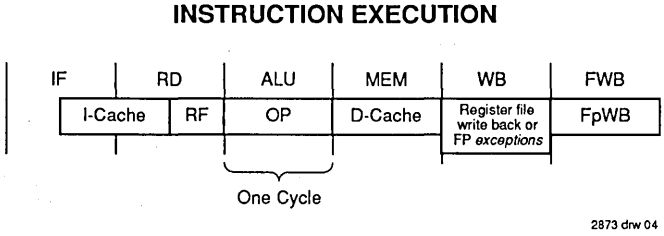


Figure 4. IDT79R3010A Instruction Summary

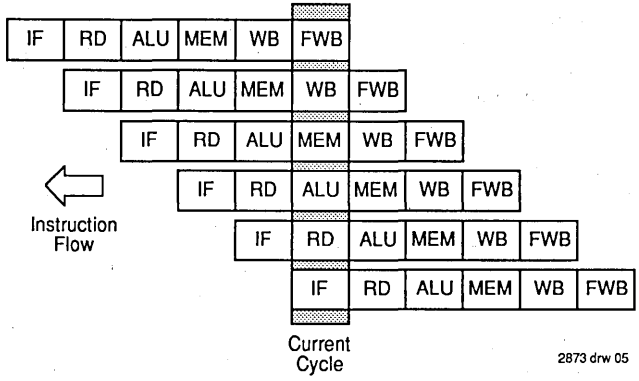


Figure 5. IDT79R3010A Instruction Pipeline

The IDT79R3010A uses a 6-stage pipeline to achieve an instruction execution rate approaching one instruction per FPA cycle. Thus, execution of six instructions at a time are overlapped as shown in Figure 5.

This pipeline operates efficiently because different FPA resources (address and data bus accesses, ALU operations, register accesses, and so on) are utilized on a non-interfering basis.

PACKAGE THERMAL SPECIFICATIONS

The IDT79R3010A utilizes special packaging techniques to improve both the thermal and electrical characteristics of the floating point accelerator.

In order to improve the electrical characteristics of the device, the package is constructed using multiple signal planes, including individual power planes and ground planes to reduce noise associated with high-frequency TTL parts.

In order to improve the thermal characteristics of the floating point accelerator, the device is housed using cavity down packaging for the flatpack and the PGA (the J-bend CerQuad is cavity up). In addition, these packages incorporate a copper-tungsten thermal slug designed to efficiently transfer heat from the die to the case of the package, and thus effectively lower the thermal resistance of the package. The use of an additional external heat sink affixed to the package thermal slug further decreases the effective thermal resistance of the package.

The case temperature may be measured in any environment to determine whether the device is within the specified operating range. The case temperature should be measured at the center of the top surface opposite the package cavity (the package cavity is the side where the package lid is mounted).

The equivalent allowable ambient temperature, T_A, can be calculated using the thermal resistance from case to ambient ($\varnothing ca$) for the given package. The following equation relates ambient and case temperature:

$$T_A = T_C - P * \varnothing ca$$

where P is the maximum power consumption, calculated by using the maximum Icc from the DC Electrical Characteristic section.

Typical values for $\varnothing ca$ at various airflows are shown in Table 2 for the various CPU packages.

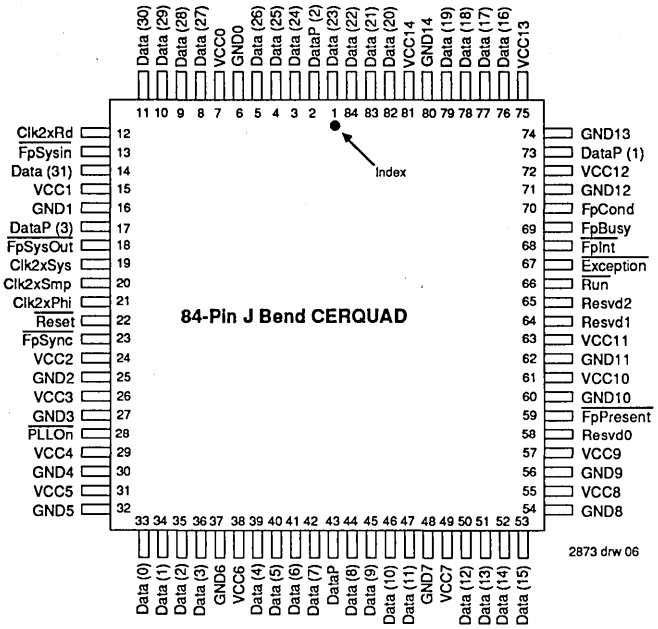
	Airflow - (ft/min)					
	0	200	400	600	800	1000
$\varnothing ca$ (84-PGA)	22	8	3	2	1.5	1.0
$\varnothing ca$ (84-Flatpack)	22	9	4	3	2	1.5
$\varnothing ca$ (84-CerQuad)	25	17	12	8	7	6

2873 tbl 02

Table 2. Thermal Resistance (Oca) at Various Airflows

5

PIN CONFIGURATION[1]
(Top View)



NOTE:
1. Reserved pins must not be connected.

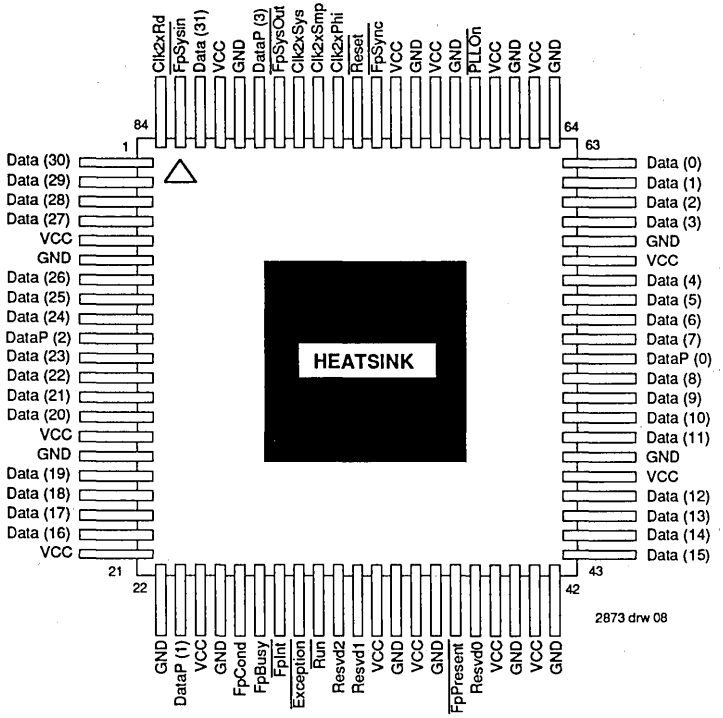
PIN CONFIGURATION[1]
(Ceramic, Cavity Down) – BOTTOM VIEW

	1	2	3	4	5	6	7	8	9	10	11	12
M	Vss	Vcc	Data 17	DataP 1	Vss	FP Cond	$\overline{\text{FPInt}}$	Vss	$\overline{\text{Run}}$	Rsrvd 1	Vcc	Vss
L	Data 21	Data 20	Data 18	Data 16	Vcc	FPBusy	Excep-tion	Vcc	Rsrvd 2	$\overline{\text{FP}}$ Present	Data 15	Data 14
K	Vss	Vcc	Data 19							Rsrvd 0	Vcc	Vss
J	Data 23	Data 22									Data 13	Data 12
H	Data 24	DataP 2									Data 11	Data 10
G	Data 26	Data 25				84-Pin Ceramic Pin Grid Array					Vcc	Vss
F	Vss	Vcc									Data 8	Data 9
E	Data 27	Data 28									Data 7	DataP 0
D	Data 29	Data 30									Data 5	Data 6
C	Vss	Vcc	Clk2x Rd							Data 2	Vcc	Vss
B	$\overline{\text{Fp}}$ SysIn	Data 31	DataP 3	Vcc	Clk2x Sys	Vcc	Clk2x Phi	Vcc	$\overline{\text{PllOn}}$	Data 1	Data 3	Data 4
A	Vss	Vcc	$\overline{\text{FpSys}}$ Out	Vss	Clk2x Smp	Vss	$\overline{\text{Reset}}$	Vss	$\overline{\text{FP}}$ Sync	Data 0	Vcc	Vss

2873 drw 07

NOTE:
1. Reserved pins must not be connected.

PIN CONFIGURATION[1]
84–L QUAD FLATPACK (CAVITY DOWN)
TOP VIEW



NOTE:
1. Reserved pins must not be connected.

PIN DESCRIPTIONS

Pin Name	I/O	Description
Data (0-31)	I/O	A multiplexed 32-bit bus used for instruction and data transfers on phase 1 and phase 2, respectively.
DataP (0-3)	O	A 4-bit bus containing even parity over the data bus. Parity is generated by the FPA on stores.
$\overline{\text{Run}}$	I	Input to the FPA which indicates whether the processor-coprocessor system is in the run or stall state.
$\overline{\text{Exception}}$	I	Input to the FPA which indicates exception related status information.
FpBusy	O	Signal to the CPU indicating a request for a coprocessor busy stall.
FpCond	O	Signal to the CPU indicating the result of the last comparision operation.
$\overline{\text{FpInt}}$	O	Signal to the CPU indicating that a floating-point exception has occured for the current FPA instruction.
$\overline{\text{Reset}}$	I	Synchronous initialization input used to distinguish the processor-FPA synchronization period from the execution period. $\overline{\text{Reset}}$ must be synchronized by the leading edge of $\overline{\text{SysOut}}$ from the CPU.
$\overline{\text{PllOn}}$	I	Input which during the reset period determines whether the phase lock mechanism is enabled and during the execution period determines the output timing model.
$\overline{\text{FpPresent}}$	O	Output which is pulled to ground through an impedance of approximately 0.5k ohms. By providing an external pullup on this line, an indication of the presence or absence of the FPA can be obtained.
Clk2xSys	I	A double frequency clock input used for generating $\overline{\text{FpSysOut}}$.
Clk2xSmp	I	A double frequency clock input used to determine the sample point for data coming in to the FPA.
Clk2xRd	I	A double frequency clock input used to determine the disable point for the data drivers.
Clk2xPhi	I	A double frequency clock input used to determine the position of the internal phases, phase 1 and phase 2.
$\overline{\text{FpSysOut}}$	O	Synchronization clock from the FPA.
$\overline{\text{FpSysIn}}$	I	Input used to receive the synchronization clock from the FPA.
$\overline{\text{FpSync}}$	I	Input used to receive the synchronization clock from the CPU.

2873 tbl 03

5

ABSOLUTE MAXIMUM RATINGS[1, 3]

Symbol	Rating	Commercial	Military	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70[4] (Ambient) 0 to +90[5] (Case)	–55 to +125 (Case)	°C
TBIAS	Case Temperature Under Bias	–55 to +125[4] 0 to +90[5]	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
IIN	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

2873 tbl 04

NOTE:
1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. VIN minimum = -3.0V for pulse width less than 15ns.
   VIN should not exceed VCC +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.
4. 16-33 MHz only.
5. 37-40 MHz only.

AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V
VIHC	Input HIGH Voltage	4.0	—	V
VILC	Input LOW Voltage	—	0.4	V

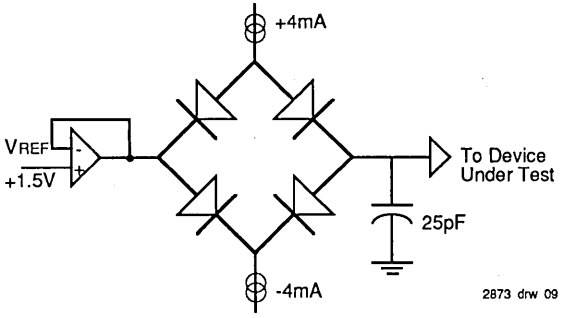
2873 tbl 05

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	Vcc
Military	-55°C to +125°C (Case)	0V	5.0 ±10%
Commercial 16-33 MHz	0°C to +70°C (Ambient)	0V	5.0 ±5%
Commercial 37-40 MHz	0°C to +90°C (Case)	0V	5.0 ±5%

2873 tbl 06

OUTPUT LOADING FOR AC TESTING



2873 drw 09

DC ELECTRICAL CHARACTERISTICS FOR IDT79R3010A COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to + 70°C, V_{CC} = + 5.0 V ± 5%)

Symbol	Parameter	Test Conditions	16.67 MHz		20.0 MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OLFP}	Output LOW Voltage(5)	V_{CC} = Min, I_{OL} = 1.5mA	—	0.5	—	0.5	V
V_{IH}	Input HIGH Voltage(6)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input High Voltage(2,6)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
V_{IHC}	Input HIGH Voltage(4,6)		4.0	—	4.0	—	V
V_{ILC}	Input LOW Voltage(1,4)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(7)		—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5.0V, T_A = 70°C	—	525	—	600	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	–100	—	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–100	100	–100	100	µA

2873 tbl 07

5

DC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to + 70°C, V_{CC} = + 5.0 V ± 5%)

Symbol	Parameter	Test Conditions	25.0 MHz		33.33 MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OLFP}	Output LOW Voltage(5)	V_{CC} = Min, I_{OL} = 1.5mA	—	0.5	—	0.5	V
V_{IH}	Input HIGH Voltage(6)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input High Voltage(2,6)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
V_{IHC}	Input HIGH Voltage(4,6)		4.0	—	4.0	—	V
V_{ILC}	Input LOW Voltage(1,4)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(7)		—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5.0V, T_A = 70°C	—	650	—	700	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	–100	—	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–100	100	–100	100	µA

2873 tbl 08

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5V for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, $\overline{\text{FpSysin}}$, $\overline{\text{FpSync}}$ and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{IHC} and V_{ILC} apply to $\overline{\text{Run}}$, $\overline{\text{PllOn}}$ and $\overline{\text{Exception}}$.
5. V_{OLFP} applies to the $\overline{\text{FPPresent}}$ pin only.
6. V_{IH} and V_{IHS} should not be held above V_{CC} + 0.5 Volts.
7. Guaranteed by design.

DC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE
COMMERCIAL TEMPERATURE RANGE (T_C = 0°C to + 90°C, V_{CC} = + 5.0 V ± 5%)

Symbol	Parameter	Test Conditions	37 MHz		40 MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OLFP}	Output LOW Voltage(5)	V_{CC} = Min, I_{OL} = 1.5mA	—	0.5	—	0.5	V
V_{IH}	Input HIGH Voltage(6)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input High Voltage(2,6)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
V_{IHC}	Input HIGH Voltage(4,6)		4.0	—	4.0	—	V
V_{ILC}	Input LOW Voltage(1,4)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(7)		—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5.0V, T_C = 90°C	—	725	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	-100	100	-100	100	µA

2873 tbl 09

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5V for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, $\overline{\text{FpSysin}}$, $\overline{\text{FpSync}}$ and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{IHC} and V_{ILC} apply to $\overline{\text{Run}}$, $\overline{\text{PllOn}}$ and $\overline{\text{Exception}}$.
5. V_{OLFP} applies to the $\overline{\text{FPPresent}}$ pin only.
6. V_{IH} and V_{IHS} should not be held above V_{CC} + 0.5 Volts.
7. Guaranteed by design.

DC ELECTRICAL CHARACTERISTICS FOR IDT79R3010A
MILITARY TEMPERATURE RANGE (T_C = –55°C to + 125°C, V_{CC} = + 5.0 V ± 10%)

Symbol	Parameter	Test Conditions	16.67 MHz		20.0 MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OLFP}	Output LOW Voltage(5)	V_{CC} = Min, I_{OL} = 1.5mA	—	0.5	—	0.5	V
V_{IH}	Input HIGH Voltage(6)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input High Voltage(2,6)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
V_{IHC}	Input HIGH Voltage(4,6)		4.0	—	4.0	—	V
V_{ILC}	Input LOW Voltage(1,4)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(7)		—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5.0V, T_A = 70°C	—	575	—	650	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–100	100	–100	100	µA

2873 tbl 10

DC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE —
MILITARY TEMPERATURE RANGE (T_C = –55°C to + 125°C, V_{CC} = + 5.0 V ± 10%)

Symbol	Parameter	Test Conditions	25.0 MHz		33.33 MHz		Unit
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OLFP}	Output LOW Voltage(5)	V_{CC} = Min, I_{OL} = 1.5mA	—	0.5	—	0.5	V
V_{IH}	Input HIGH Voltage(6)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input High Voltage(2,6)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
V_{IHC}	Input HIGH Voltage(4,6)		4.0	—	4.0	—	V
V_{ILC}	Input LOW Voltage(1,4)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(7)		—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5.0V, T_A = 70°C	—	700	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–100	100	–100	100	µA

2873 tbl 11

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5V for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, $\overline{\text{FpSysin}}$, $\overline{\text{FpSync}}$ and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{IHC} and V_{ILC} apply to $\overline{\text{Run}}$, $\overline{\text{PllOn}}$ and $\overline{\text{Exception}}$.
5. V_{OLFP} applies to the $\overline{\text{FPPresent}}$ pin only.
6. V_{IH} and V_{IHS} should not be held above V_{CC} + 0.5 Volts.
7. Guaranteed by design.

AC ELECTRICAL CHARACTERISTICS FOR IDT79R3010A(1, 3)
COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ± 5%)

Symbol	Parameter	Test Conditions	16.67 MHz		20.0 MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
T_{CkHigh}	Input Clock High(2)	Note 7	12	—	10	—	ns
T_{CkLow}	Input Clock Low(2)	Note 7	12	—	10	—	ns
T_{CkP}	Input Clock Period		30	1000	25	1000	ns
	Clk2xSys to Clk2XSmp(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi(5)		9	tcyc/4	7	tcyc/4	ns
Timing Paramters							
T_{DEn}	Data Enable(3)		—	-2	—	-2	ns
T_{DDis}	Data Disable(3)		—	-1	—	-1	ns
T_{DVal}	Data Valid	Load= 25pF	—	3	—	3	ns
T_{RSDS}	Reset Set-up		15	—	15	—	ns
T_{DS}	Data Set-up		9	—	8	—	ns
T_{DH}	Data Hold(3)		-2.5	—	-2.5	—	ns
T_{FpCond}	Fp Condition		—	35	—	30	ns
T_{FpBusy}	Fp Busy		—	15	—	13	ns
T_{FpInt}	Fp Interrupt		—	40	—	35	ns
T_{FpMov}	Fp Move To		—	35	—	30	ns
T_{RExS}	Exception Set-up (Run Cycle)		14	—	12	—	ns
T_{SExS}	Exception Set-up (Stall Cycle)		12	—	10	—	ns
T_{ExH}	Exception Hold		0	—	0	—	ns
T_{RunS}	Run Set-up		17	—	15	—	ns
T_{RunH}	Run Hold		-2	—	-2	—	ns
T_{StallS}	Stall Set-up		10	—	10	—	ns
T_{StallH}	Stall Hold		-2	—	-2	—	ns
Reset Initialization							
T_{rstPLL}	Reset Timing, Phase-lock on(4, 5)		3000	—	3000	—	Tcyc
T_{rst}	Reset Timing, Phase-lock off(5)		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate(6)		0.5	2	0.5	1	ns/25pF

2873 tbl 12

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. With $\overline{PllOn}$ asserted, $\overline{Reset}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. T_{cyc} is one CPU clock cycle (two cycles of a 2x clock).
6. No two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 5ns.

AC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE(1, 3)
COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ± 5%)

Symbol	Parameter	Test Conditions	25.0 MHz		33.33 MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
TCkHigh	Input Clock High(2)	Note 7	8	—	6	—	ns
TCkLow	Input Clock Low(2)	Note 7	8	—	6	—	ns
TCkP	Input Clock Period		20	1000	15	1000	ns
	Clk2xSys to Clk2XSmp(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi(5)		5	tcyc/4	3.5	tcyc/4	ns
Timing Paramters							
TDEn	Data Enable(3)		—	-1.5	—	-1	ns
TDDis	Data Disable(3)		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	2	ns
TRSDS	Reset Set-up		10	—	10	—	ns
TDS	Data Set-up		6	—	4.5	—	ns
TDH	Data Hold(3)		-2.5	—	-2.5	—	ns
TFpCond	Fp Condition		—	25	—	17	ns
TFpBusy	Fp Busy		—	10	—	7	ns
TFpInt	Fp Interrupt		—	25	—	18	ns
TFpMov	Fp Move To		—	25	—	16	ns
TRExS	Exception Set-up (Run Cycle)		11	—	9	—	ns
TSExS	Exception Set-up (Stall Cycle)		8	—	6.5	—	ns
TExH	Exception Hold		0	—	0	—	ns
TRunS	Run Set-up		15	—	12.5	—	ns
TRunH	Run Hold		-2	—	-1.5	—	ns
TStallS	Stall Set-up		9	—	7	—	ns
TStallH	Stall Hold		-2	—	-2	—	ns
Reset Initialization							
TrstPLL	Reset Timing, Phase-lock on(4, 5)		3000	—	3000	—	Tcyc
Trst	Reset Timing, Phase-lock off(5)		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate(6)		0.5	1	0.5	1	ns/25pF

2873 tbl 13

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. With $\overline{\text{PllOn}}$ asserted, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. No two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33MHz; clock transition time < 5ns for all other speeds.

5

AC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE(1, 3)

COMMERCIAL TEMPERATURE RANGE (Tc = 0°C to +90°C, Vcc = +5.0V ± 5%)

Symbol	Parameter	Test Conditions	37.0 MHz		40.0 MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
TCkHigh	Input Clock High(2)	Note 7	5.5	—	5.5	—	ns
TCkLow	Input Clock Low(2)	Note 7	5.5	—	5.5	—	ns
TCkP	Input Clock Period		13.5	1000	12.5	1000	ns
	Clk2xSys to Clk2XSmp(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi(5)		3.5	tcyc/4	3	tcyc/4	ns
Timing Paramters							
TDEn	Data Enable(3)		—	-1.5	—	-1	ns
TDDis	Data Disable(3)		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	2	ns
TRSDS	Reset Set-up		8	—	8	—	ns
TDS	Data Set-up		4.5	—	4	—	ns
TDH	Data Hold(3)		-2.5	—	-2.5	—	ns
TFpCond	Fp Condition		—	17	—	16	ns
TFpBusy	Fp Busy		—	6.5	—	6	ns
TFpInt	Fp Interrupt		—	18	—	17	ns
TFpMov	Fp Move To		—	16	—	16	ns
TRExS	Exception Set-up (Run Cycle)		9	—	8.5	—	ns
TSExS	Exception Set-up (Stall Cycle)		6	—	5.5	—	ns
TExH	Exception Hold		0	—	0	—	ns
TRunS	Run Set-up		10	—	9	—	ns
TRunH	Run Hold		-2	—	-1.5	—	ns
TStallS	Stall Set-up		6.5	—	6	—	ns
TStallH	Stall Hold		-2	—	-2	—	ns
Reset Initialization							
TrstPLL	Reset Timing, Phase-lock on(4, 5)		3000	—	3000	—	Tcyc
Trst	Reset Timing, Phase-lock off(5)		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate(6)		0.5	1	0.5	1	ns/25pF

2873 tbl 14

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. With $\overline{\text{PllOn}}$ asserted, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. No two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns.

AC ELECTRICAL CHARACTERISTICS FOR IDT79R3010A(1, 3)
MILITARY TEMPERATURE RANGE (Tc = –55°C to +125°C, Vcc = +5.0V ± 10%)

Symbol	Parameter	Test Conditions	16.67 MHz		20.0 MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
TCkHigh	Input Clock High(2)	Note 7	12	—	10	—	ns
TCkLow	Input Clock Low(2)	Note 7	12	—	10	—	ns
TCkP	Input Clock Period		30	1000	25	1000	ns
	Clk2xSys to Clk2XSmp(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi(5)		9	tcyc/4	7	tcyc/4	ns
Timing Paramters							
TDEn	Data Enable(3)		—	-2	—	-2	ns
TDDis	Data Disable(3)		—	-1	—	-1	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	ns
TRSDS	Reset Set-up		15	—	15	—	ns
TDS	Data Set-up		9	—	8	—	ns
TDH	Data Hold(3)		-2.5	—	-2.5	—	ns
TFpCond	Fp Condition		—	35	—	30	ns
TFpBusy	Fp Busy		—	15	—	13	ns
TFpInt	Fp Interrupt		—	40	—	35	ns
TFpMov	Fp Move To		—	35	—	30	ns
TRExS	Exception Set-up (Run Cycle)		14	—	12	—	ns
TSExS	Exception Set-up (Stall Cycle)		12	—	10	—	ns
TExH	Exception Hold		0	—	0	—	ns
TRunS	Run Set-up		17	—	15	—	ns
TRunH	Run Hold		-2	—	-2	—	ns
TStallS	Stall Set-up		10	—	10	—	ns
TStallH	Stall Hold		-2	—	-2	—	ns
Reset Initialization							
TrstPLL	Reset Timing, Phase-lock on(4, 5)		3000	—	3000	—	Tcyc
Trst	Reset Timing, Phase-lock off(5)		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate(6)		0.5	2	0.5	2	ns/25pF

2873 tbl 15

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. With $\overline{\text{PllOn}}$ asserted, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. No two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 5ns

AC ELECTRICAL CHARACTERISTICS FOR IDT79R3010AE(1, 3)
MILITARY TEMPERATURE RANGE (Tc = –55°C to +125°C, Vcc = +5.0V ± 10%)

Symbol	Parameter	Test Conditions	25.0 MHz		33.33 MHz		Unit
			Min.	Max.	Min.	Max.	
Clock							
TCkHigh	Input Clock High(2)	Note 7	8	—	6	—	ns
TCkLow	Input Clock Low(2)	Note 7	8	—	6	—	ns
TCkP	Input Clock Period		20	1000	15	1000	ns
	Clk2xSys to Clk2XSmp(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd(5)		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi(5)		5	tcyc/4	3.5	tcyc/4	ns
Timing Paramters							
TDEn	Data Enable(3)		—	-1.5	—	-1	ns
TDDIs	Data Disable(3)		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	2	ns
TRSDS	Reset Set-up		10	—	10	—	ns
TDS	Data Set-up		6	—	4.5	—	ns
TDH	Data Hold(3)		-2.5	—	-2.5	—	ns
TFpCond	Fp Condition		—	25	—	17	ns
TFpBusy	Fp Busy		—	10	—	7	ns
TFpInt	Fp Interrupt		—	25	—	18	ns
TFpMov	Fp Move To		—	25	—	16	ns
TRExS	Exception Set-up (Run Cycle)		11	—	9	—	ns
TSExS	Exception Set-up (Stall Cycle)		8	—	6.5	—	ns
TExH	Exception Hold		0	—	0	—	ns
TRunS	Run Set-up		15	—	12.5	—	ns
TRunH	Run Hold		-2	—	-1.5	—	ns
TStallS	Stall Set-up		9	—	7	—	ns
TStallH	Stall Hold		-2	—	-2	—	ns
Reset Initialization							
TrstPLL	Reset Timing, Phase-lock on(4, 5)		3000	—	3000	—	Tcyc
Trst	Reset Timing, Phase-lock off(5)		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate(6)		0.5	1	0.5	1	ns/25pF

2873 tbl 15

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. With $\overline{\text{PllOn}}$ asserted, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. No two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33MHz; clock transition time < 5ns for all other speeds.

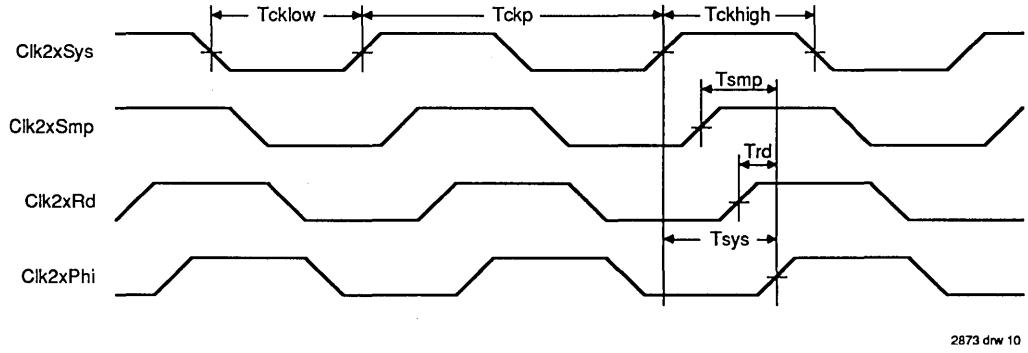


Figure 6. Input "2x" Clock Timing

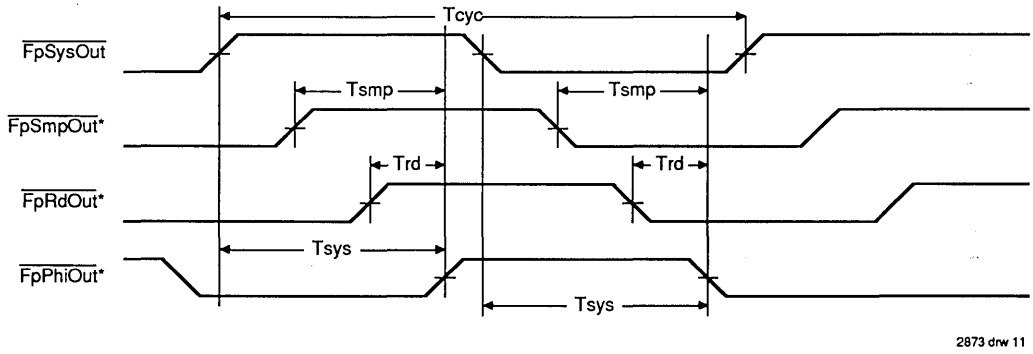


Figure 7. Processor Reference Clock

* These signals are not actually output from the floating point processor.
They are drawn to provide a reference for other timing diagrams.

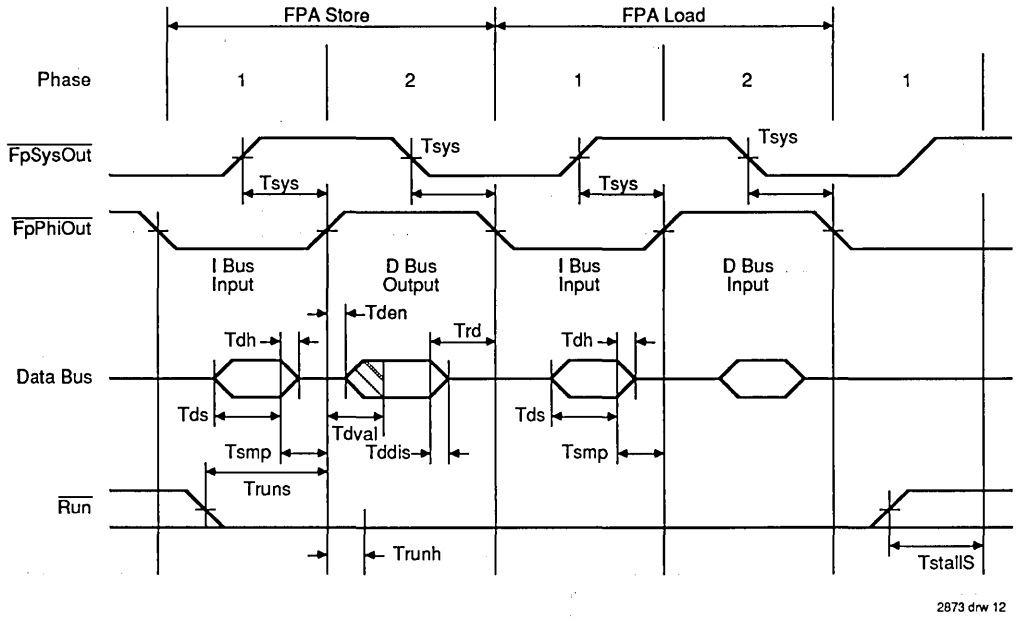


Figure 8. Floating Point Load/Store Timing

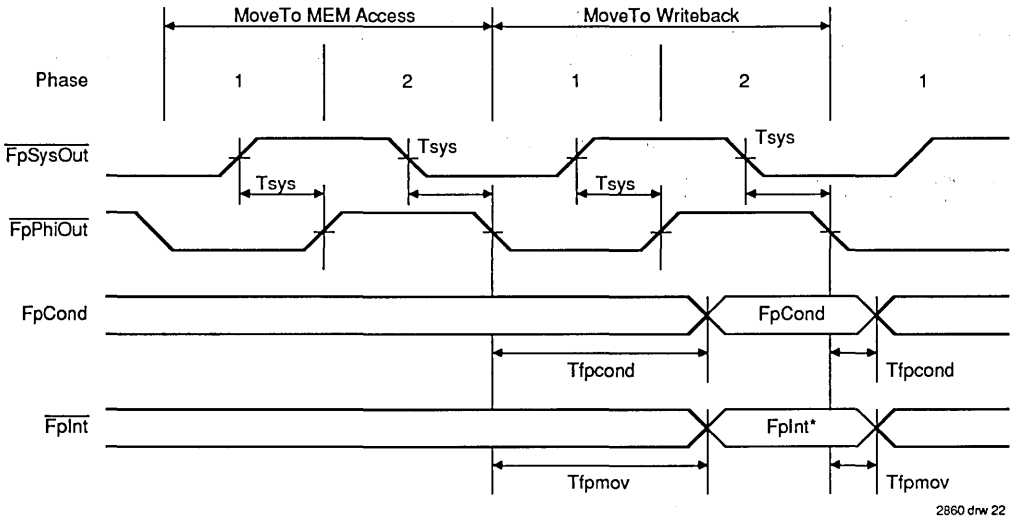


Figure 9. Move to FPC Status Timing

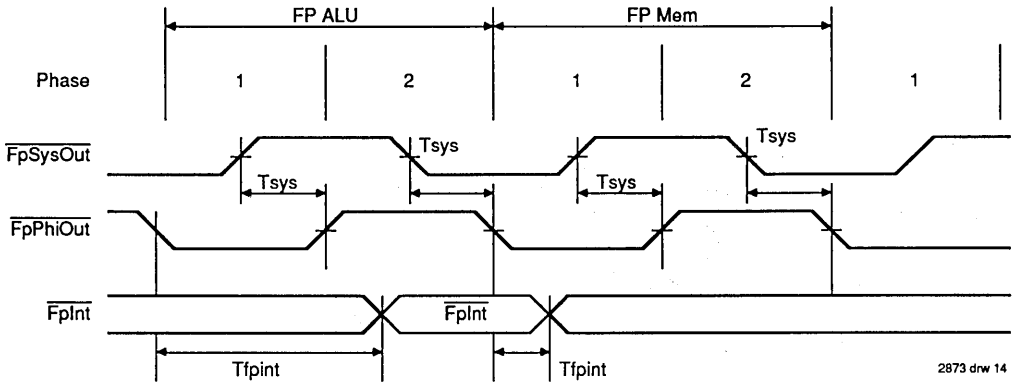


Figure 10. Floating Point Interrupt Timing

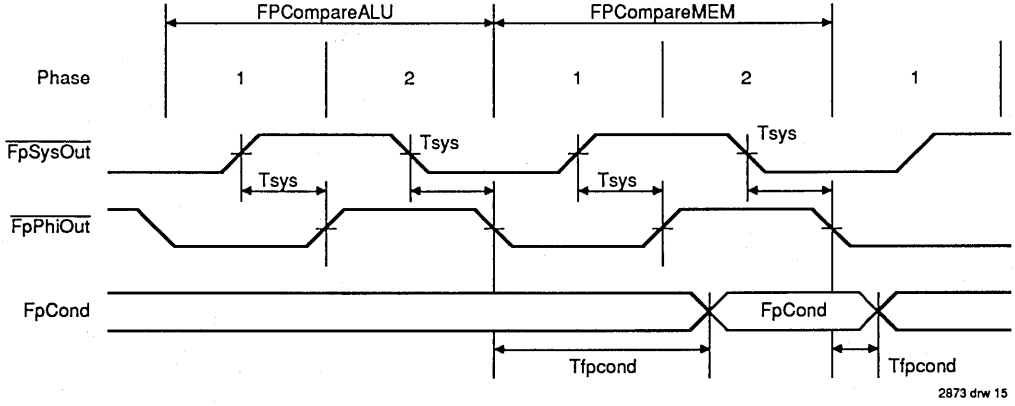


Figure 11. Floating Point Condition Timing

5

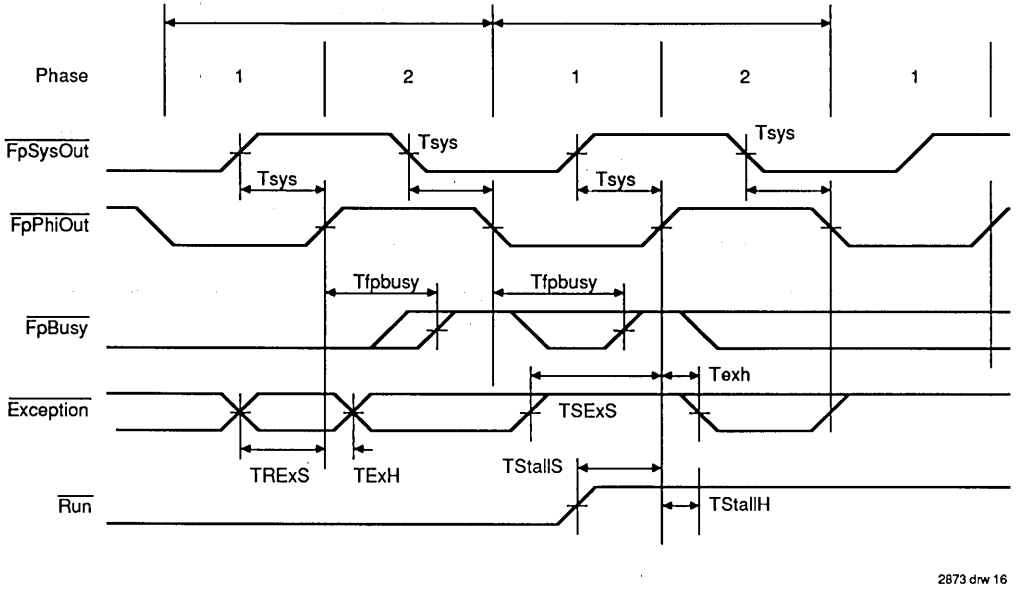


Figure 12. Floating Point Busy, Exception Timing

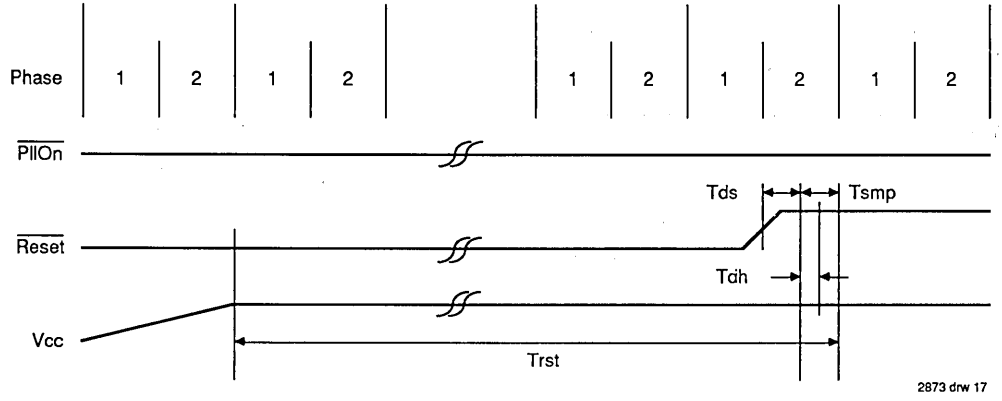
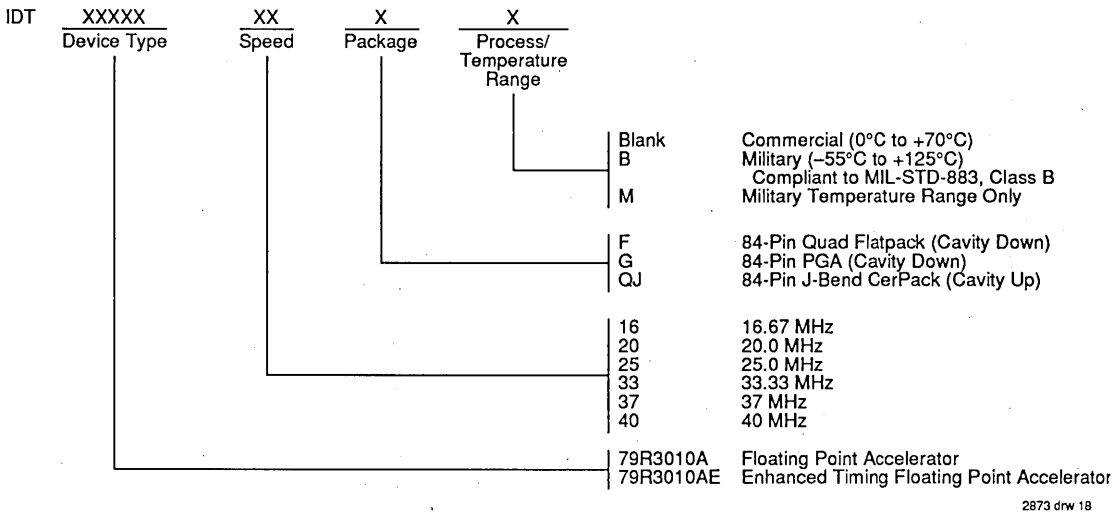


Figure 13. Power-On Reset Timing

ORDERING INFORMATION



2873 drw 18

5

Integrated Device Technology, Inc.

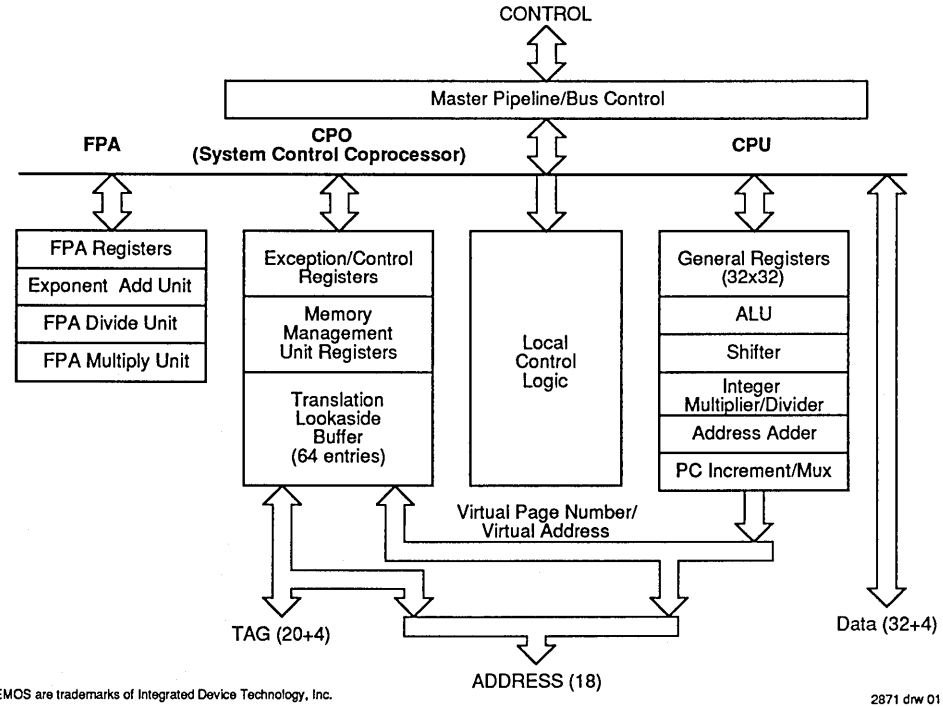
RISCore™ RISC CPU PROCESSOR

PRELIMINARY
IDT79R3500A

FEATURES:

- A single chip integrating the R3000 CPU and R3010 FPA execution units, using the R3000A pinout.
- Efficient Pipelining—The CPU's 5-stage pipeline design assists in obtaining an execution rate approaching one instruction per cycle. Pipeline stalls and exceptions are handled precisely and efficiently.
- On-Chip Cache Control—The IDT79R3500A provides a high bandwidth memory interface that handles separate external Instruction and Data Caches ranging in size from 4 to 256 Kbytes each. Both caches are accessed during a single CPU cycle. All cache control is on-chip.
- On-Chip Memory Management Unit—A fully-associative, 64 entry Translation Lookaside Buffer (TLB) provides fast address translation for virtual-to-physical memory mapping of the 4 Gigabyte virtual address space.
- Dynamically able to switch between Big- and Little- Endian byte ordering conventions.
- Optimizing Compilers are available for C, FORTRAN, Pascal, COBOL, Ada, and PL/1.
- 16.7 through 40MHz clock rates yield up to 32 VUPS sustained throughput.
- Supports independent multiword block refill of both the instruction and data caches with variable block sizes.
- Supports concurrent refill and execution of instructions.
- Partial word stores executed as read-modify-write.
- 6 external interrupt inputs, 2 software interrupts, with single cycle latency to exception handler routine.
- Flexible multiprocessing support on chip with no impact on uniprocessor designs.
- Software compatible with R3000, R2000 CPUs and R3010, 2010 FPAs.
- Faster integer multiply and divide than standard R3000
- TLB disable feature allowing a simple memory model for Embedded Applications.
- Programmable Tag bus width allowing reduced cost cache.
- Hardware Support of Single- and Double-Precision Floating Point Operations that include Add, Subtract, Multiply, Divide, Comparisons, and Conversions.
- Sustained Floating Point Performance of 11 MFlops single precision LINPACK and 7.3 MFLOPS double precision
- Supports Full Conformance With IEEE 754-1985 Floating Point Specification
- 64-bit FP operation using sixteen 64-bit data registers

IDT79R3500A PROCESSOR



2871 drw 01

DESCRIPTION:

The IDT 79R3500A RISC Microprocessor consists of three tightly-coupled processors integrated on a single chip. The first processor is a full 32-bit CPU based on RISC (Reduced Instruction Set Computer) principles to achieve a new standard of microprocessor performance. The second processor is a system control coprocessor, called CP0, containing a fully-associative 64 entry TLB (Translation Lookaside Buffer), MMU (Memory Management Unit) and control registers, supporting a 4 Gigabyte virtual memory subsystem, and a Harvard Architecture Cache Controller achieving a bandwidth of 320 Mbytes/second using industry standard static RAMs. The third processor is the Floating Point Accelerator which performs arithmetic operations on values in floating-point representations. This processor fully conforms to the requirements of ANSI/IEEE Standard 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic." In addition, the architecture fully supports the standard's recommendations.

The programmer model of this device will be the same as the programmer model of a system which uses a discrete 79R3000 with the 79R3010: 32 integer registers, 16 floating point registers; co-processor 0 registers; floating point status and control register; RISC integer ALU; Integer Multiply and Divide ALU; Floating Point Add/Subtract, Multiply, and Divide ALUs. The device pipeline will be the same as for the 79R3000, as will the co-processor 0 functionality. No new instructions have been introduced. Pin compatibility extends to AC and DC characteristics, software execution and initialization mode vector selection.

This data sheet provides an overview of the features and architecture of the 79R3500A CPU, Revision 3.0. A more detailed description of the operation of the device is incorporated in the "R3500A Family Hardware User Manual", and a more detailed architectural overview is provided in the "mips RISC Architecture" book, both available from IDT. Documentation providing details of the software and development environments supporting this processor are also available from IDT.

IDT79R3500A CPU Registers

The IDT79R3500A CPU provides 32 general purpose 32-bit registers, a 32-bit Program Counter, and two 32-bit registers that hold the results of integer multiply and divide operations. Only two of the 32 general registers have a special purpose: register r0 is hardwired to the value "0", which is a useful constant, and register r31 is used as the link register in jump-and-link instructions (return address for subroutine calls).

The CPU registers are shown in Figure 2. Note that there is no Program Status Word (PSW) register shown in this figure: the functions traditionally provided by a PSW register are instead provided in the Status and Cause registers incorporated within the System Control Coprocessor (CP0).

FPA REGISTERS

The IDT79R3010A FPA provides 32 general purpose 32-bit registers, a Control/Status register, and a Revision Identification register.

Floating-point coprocessor operations reference three types of registers:

- Floating-Point Control Registers (FCR)
- Floating-Point General Registers (FGR)
- Floating-Point Registers (FPR)

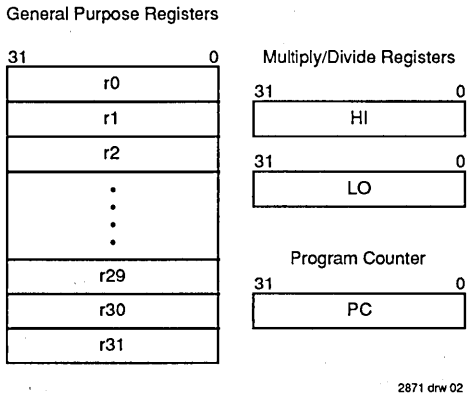


Figure 2. IDT79R3500A CPU Registers

Floating-Point General Registers (FGR)

There are 32 Floating-Point General Registers (FGR) on the FPA. They represent directly-addressable 32-bit registers, and can be accessed by Load, Store, or Move Operations.

Floating-Point Registers (FPR)

The 32 FGRs described in the preceding paragraph are also used to form sixteen 64-bit Floating-Point Registers (FPR). Pairs of general registers (FGRs), for example FGR0 and FGR1 (refer to Figure 3) are physically combined to form a single 64-bit FPR. The FPRs hold a value in either single- or double-precision floating-point format. Double-precision format FPRs are formed from two adjacent FGRs.

Floating-Point Control Registers (FCR)

There are 2 Floating-Point Control Registers (FCR) on the FPA. They can be accessed only by Move operations and include the following:

- Control/Status register, used to control and monitor exceptions, operating modes, and rounding modes;
- Revision register, containing revision information about the FPA.

5

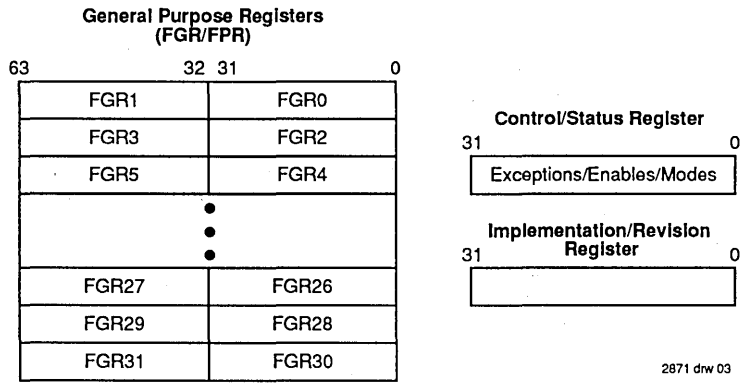


Figure 3. FPA Registers

Instruction Set Overview

All IDT79R3500A instructions are 32 bits long, and there are only three instruction formats. This approach simplifies instruction decoding, thus minimizing instruction execution time. The 79R3500A processor initiates a new instruction on every run cycle, and is able to complete an instruction on almost every clock cycle. The only exceptions are the Load instructions and Branch instructions, which each have a single cycle of latency associated with their execution. Note, however, that in the majority of cases the compilers are able to fill these latency cycles with useful instructions which do not require the result of the previous instruction. This effectively eliminates these latency effects.

The actual instruction set of the CPU was determined after extensive simulations to determine which instructions should be implemented in hardware, and which operations are best synthesized in software from other basic instructions. This methodology resulted in the R3500A having the highest performance of any available microprocessor.

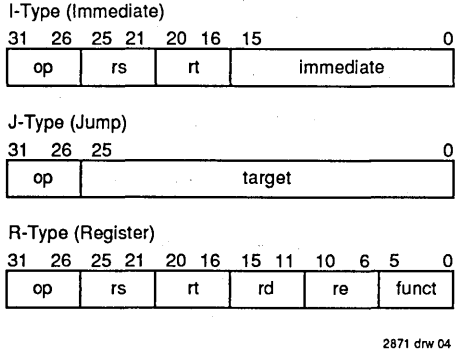


Figure 4. IDT79R3500A Instruction Formats

The IDT79R3500A instruction set can be divided into the following groups:

- **Load/Store** instructions move data between memory and general registers. They are all I-type instructions, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.
  The Load instruction has a single cycle of latency, which means that the data being loaded is not available to the instruction immediately after the load instruction. The compiler will fill this delay slot with either an instruction which is not dependent on the loaded data, or with a NOP instruction. There is no latency associated with the store instruction.
  Loads and Stores can be performed on byte, half-word, word, or unaligned word data (32 bit data not aligned on a modulo-4 address). The CPU cache is constructed as a write-through cache.
- **Computational** instructions perform arithmetic, logical and shift operations on values in registers. They occur in both R-type (both operands and the result are registers) and I-type (one operand is a 16-bit immediate) formats. FP computational instructions perform arithmetic operations on floating point values in the FPA registers. Note that computational instructions are three operand instructions; that is, the result of the operation can be stored into a different register than either of the two operands. This means that operands need not be overwritten by arithmetic operations. This results in a more efficient use of the large register set.
- **Conversion** instructions perform conversion operations on the floating point values in the FPA registers.
- **Compare** intructions perform comparisons of the contents of FPA registers and set a condition bit based on the results. The result of the compare operations is tied directly to Cp Cond (1) for software testing.
- **Jump and Branch** instructions change the control flow of a program. Jumps are always to a paged absolute address formed by combining a 26-bit target with four bits of the Program counter (J-type format, for subroutine calls), or

OP	Description	OP	Description
	Load/Store Instructions		**Shift Instructions (Cont.)**
LB	Load Byte	SRA	Shift Right Arithmetic
LBU	Load Byte Unsigned	SLLV	Shift Left Logical Variable
LH	Load Halfword	SRLV	Shift Right Logical Variable
LHU	Load Halfword Unsigned	SRAV	Shift Right Arithmetic Variable
LW	Load Word		
LWL	Load Word Left		**FPA Conversion Instructions**
LWR	Load Word Right	CVT.S.fmt	Floating point Convert to Single FP
SB	Store Byte	CVT.D.fmt	Floating point Convert to Double FP
SH	Store Halfword	CVT.W.fmt	Floating point Convert to fixed point
SW	Store Word		
SWL	Store Word Left		**Multiply/Divide Instructions**
SWR	Store Word Right	MULT	Multiply
		MULTU	Multiply Unsigned
	FPA Load/Store/Move Instructions	DIV	Divide
LWC1	Load Word to FPA	DIVU	Divide Unsigned
SWC1	Store Word from FPA	MFHI	Move From HI
MTC1	Move Word to FPA	MTHI	Move To HI
MFC1	Move Word from FPA	MFLO	Move From LO
CTC1	Move Control word to FPA	MTLO	Move To LO
CFC1	Move Control word from FPA		
			Jump and Branch Instructions
	Arithmetic Instructions (ALU Immediate)	J	Jump
		JAL	Jump and Link
ADDI	Add Immediate	JR	Jump to Register
ADDIU	Add Immediate Unsigned	JALR	Jump and Link Register
SLTI	Set on Less Than Immediate	BEQ	Branch on Equal
SLTIU	Set on Less Than Immediate Unsigned	BNE	Branch on Not Equal
		BLEZ	Branch on Less than or Equal to Zero
ANDI	AND Immediate	BGTZ	Branch on Greater Than Zero
ORI	OR Immediate	BLTZ	Branch on Less Than Zero
XORI	Exclusive OR Immediate	BGEZ	Branch on Greater than or Equal to Zero
LUI	Load Upper Immediate		
		BLTZAL	Branch on Less Than Zero and Link
	Arithmetic Instructions (3-operand, register-type)	BGEZAL	Branch on Greater than or Equal to Zero and Link
ADD	Add		
ADDU	Add Unsigned		**Special Instructions**
SUB	Subtract	SYSCALL	System Call
SUBU	Subtract Unsigned	BREAK	Break
SLT	Set on Less Than		
SLTU	Set on Less Than Unsigned		**Coprocessor Instructions**
AND	AND	LWCZ	Load Word from Coprocessor
OR	OR	SWCZ	Store Word to Coprocessor
XOR	Exclusive OR	MTCZ	Move To Coprocessor
NOR	NOR	MFCZ	Move From Coprocessor
		CTCZ	Move Control to Coprocessor
	FPA Computational Instructions	CFCZ	Move Control From Coprocessor
ADD.fmt	Floating point Add	COPZ	Coprocessor Operation
SUB.fmt	Floating point Subtract	BCZT	Branch on Coprocessor z True
MUL.fmt	Floating point Multiply	BCZF	Branch on Coprocessor z False
DIV.fmt	Floating point Divide		
ABS.fmt	Floating-point Absolute value		**System Control Coprocessor (CP0) Instructions**
MOV.fmt	Floating point Move		
NEG.fmt	Floating point Negate	MTC0	Move To CP0
		MFC0	Move From CP0
	FPA Compare Instructions	TLBR	Read indexed TLB entry
C.cond.fmt	Floating-point Compare	TLBWI	Write Indexed TLB entry
		TLBWR	Write Random TLB entry
	Shift Instructions	TLBP	Probe TLB for matching entry
SLL	Shift Left Logical	RFE	Restore From Exception
SRL	Shift Right Logical		

Table 1. IDT79R3500A Instruction Summary

2871 tbl 01

5

32-bit register byte addresses (R-type, for returns and dispatches). Branches have 16-bit offsets relative to the program counter (I-type). Jump and Link instructions save a return address in Register 31. The 79R3500A instruction set features a number of branch conditions. Included is the ability to compare a register to zero and branch, and also the ability to branch based on a comparison between two registers. Thus, net performance is increased since software does not have to perform arithmetic instructions prior to the branch to set up the branch conditions.

- **Coprocessor** instructions perform operations in the coprocessors. Coprocessor Loads and Stores are I-type.
- **Coprocessor 0** instructions perform operations on the System Control Coprocessor (CP0) registers to manipulate the memory management and exception handling facilities of the processor.
- **Special** instructions perform a variety of tasks, including movement of data between special and general registers, system calls, and breakpoint. They are always R-type.

Table 1 lists the instruction set of the IDT79R3500A processor.

IDT79R3500A System Control Coprocessor (CP0)

The IDT79R3500A can operate with up to four tightly-coupled coprocessors (designated CP0 through CP3). The System Control Coprocessor (or CP0), is incorporated on the IDT79R3500A chip and supports the virtual memory system and exception handling functions of the IDT79R3500A. The virtual memory system is implemented using a Translation Lookaside Buffer and a group of programmable registers as shown in Figure 5.

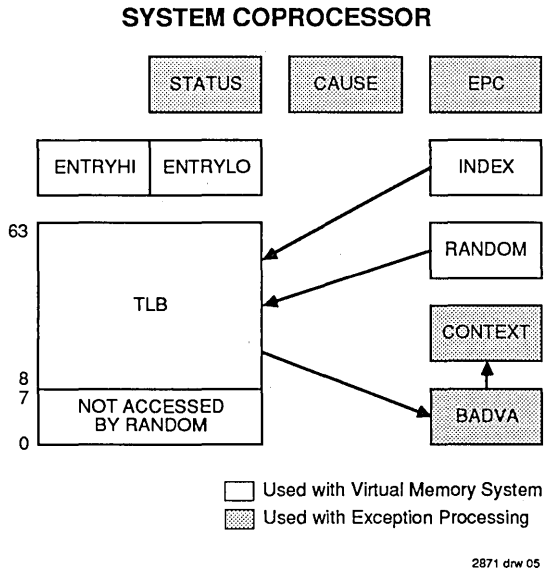
System Control Coprocessor (CP0) Registers

The CP0 registers shown in Figure 5 are used to control the memory management and exception handling capabilities of the IDT79R3500A. Table 2 provides a brief description of each register.

Register	Description
EntryHi	High half of a TLB entry
EntryLo	Low half of a TLB entry
Index	Programmable pointer into TLB array
Random	Pseudo-random pointer into TLB array
Status	Mode, interrupt enables, and diagnostic status info
Cause	Indicates nature of last exception
EPC	Exception Program Counter
Context	Pointer into kernel's virtual Page Table Entry array
BadVA	Most recent bad virtual address
PRId	Processor revision identification (Read only)

2871 tbl 02

Table 2. System Control Coprocessor (CP0) Registers



2871 drw 05

Figure 5. The System Coprocessor Registers

Memory Management System

The IDT79R3500A has an addressing range of 4 Gbytes. However, since most IDT79R3500A systems implement a physical memory smaller than 4Gbytes, the IDT79R3500A provides for the logical expansion of memory space by translating addresses composed in a large virtual address space into available physical memory address. Two TLB modes are supported. When the TLB is used, the 4 GByte address space is divided into 2 GBytes which can be accessed by both the users and the kernel, and 2 GBytes for the kernel only. Virtual addresses within the kernel/user segment are translated to physical addresses on a 4kB page basis. This mode is typical of UNIX and other sophisticated operating systems. When the TLB is disabled, mapping is locked as 2 GBytes as kernel/user, and 1.5 GBytes as kernel only. This mode requires no TLB manipulation, provides large linear address space, and is typical for embedded applications.

TLB (Translation Lookaside Buffer)

Virtual memory mapping is assisted by the Translation Lookaside Buffer (TLB). The on-chip TLB provides very fast virtual memory access and is well-matched to the requirements of multi-tasking operating systems. The fully-associative TLB contains 64 entries, each of which maps a 4-Kbyte page, with controls for read/write access, cacheability, and process identification. The TLB allows each user to access up to 2 Gbytes of virtual address space.

Figure 6 illustrates the format of each TLB entry. The Translation operation involves matching the current Process ID (PID) and upper 20 bits of the address against PID and VPN (Virtual Page Number) fields in the TLB. When both match (or the TLB entry is Global), the VPN is replaced with the PFN (Physical Frame Number) to form the physical address.

TLB misses are handled in software, with the entry to be replaced determined by as imple RANDOM function. The routine to process a TLB miss in the UNIX environment requires only 10-12 cycles, which compares favorably with many CPUs which perform the operation in hardware.

TLB Disabled Operation

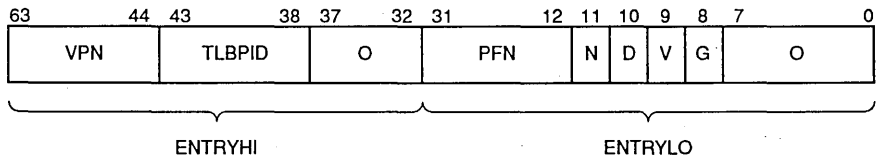
Many embedded systems do not like the complexity or uncertainty associated with the on-chip TLB. However, many systems still desire the ability to implement a kernel/user mode. Therefore, to implement a hierachical task model, the TLB must be used. The IDTR3500A gives the system designer one more option, allowing the TLB to be disabled and performing a fixed mapping of virtual to physical addresses, while maintaining separation of kernel and user resources.

The user may elect to disable the TLB through the reset sectors. In this case, the mapping shown in Figure 8. is used, and device power consumption is reduced. Note that "cached" segments means that there is no mechanism to exclude addresses in these regions from the cache.

This mapping means that applications designed to run in kseg0 and kseg1 (to avoid the TLB) can use the IDT3500A, disable the TLB to reduce power, and not have to change software to take advantage of this new feature.

5

TLB ENTRY FORMAT



VPN – Virtual Page Number
TLBPID – Process ID
PFN – Physical Frame Number
N – Non-cacheable flag
D – Dirty flag (Write protect)
V – Valid entry flag
G – Global flag (ignore PID)
O – Reserved

2871 drw 06

Figure 6. TLB Entry Format

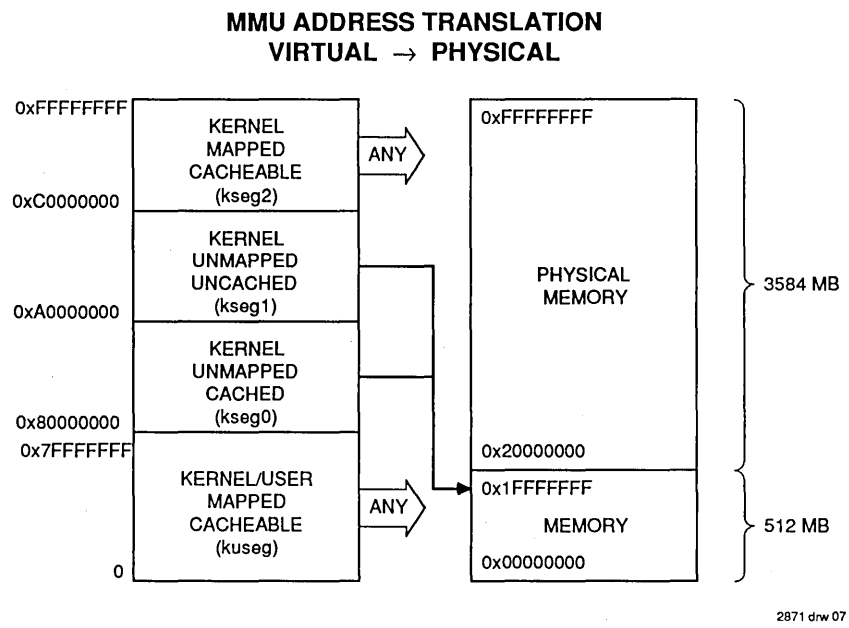


Figure 7. IDT79R3500A Virtual Address Mapping

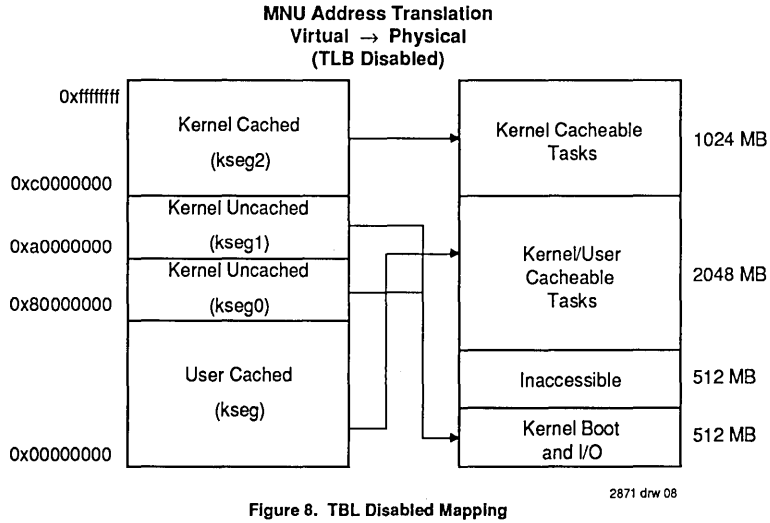


Figure 8. TBL Disabled Mapping

NOTE: This model is consistent with the mapping available in the IDT79R3051 family. The identical mapping provides software compatibility to the lower cost CPUs.

Operating Modes

The IDT79R3500A has two operating modes: User mode and Kernel mode. The IDT79R3500A normally operates in the User mode until an exception is detected forcing it into the Kernel mode. It remains in the Kernel mode until a Restore From Exception (RFE) instruction is executed. The manner in which memory addresses are translated or mapped depends on the operating mode of the IDT79R3500A. Figure 7 shows the MMU translation performed for each of the operating modes.

User Mode—in this mode, a single, uniform virtual address space (kuseg) of 2 Gbyte is available. When the TLB is used, each virtual address is extended with a 6-bit process identifier field to form unique virtual addresses. All references to this segment are mapped through the TLB. Use of the cache for up to 64 processes is determined by bit settings for each page within the TLB entries. If the TLB is not used, these addresses are translated to begin at 1Gbyte of the physical address space.

Kernel Mode—four separate segments are defined in this mode:

- *kuseg*—when in the kernel mode, references to this segment are treated just like user mode references, thus streamlining kernel access to user data.
- *kseg0*—references to this 512 Mbyte segment use cache memory but are not mapped through the TLB. Instead, they always map to the first 0.5 GBytes of physical address space.
- *kseg1*—references to this 512 Mbyte segment are not mapped through the TLB and do not use the cache. Instead, they are hard-mapped into the same 0.5 GByte segment of physical address space as kseg0.
- *kseg2*—when the TLB is not used, references to this 1Gbyte segment directly addresses the upper 1Gbyte of physical address space. These addresses are defined to be kernel mode which are cacheable. When the TLB is used, references to this 1Gbyte segment are always mapped through the TLB and use of the cache is determined by bit settings within the TLB entry.

FPA COPROCESSOR OPERATION (CP1)

The FPA continually monitors the processor instruction stream. If an instruction does not apply to the coprocessor, it is ignored; if an instruction does apply to the coprocessor, the FPA executes that instruction and transfers necessary result and exception data synchronously to the main processor.

The FPA performs three types of operations:

- Loads and Stores;
- Moves;
- Two- and three-register floating-point operations.

Load, Store, and Move Operation

Load, Store, and Move operations data between memory or the integer registers and the FPA registers. These operations perform no format conversions and cause no floating-point exceptions. Load, Store, and Move operations reference a single 32-bit word of either the Floating-Point General Registers (FGR) or the Floating-Point Control Registers (FCR).

Floating-Point Operations

The FPA supports the following single- and double-precision format floating-point operations:

- Add
- Subtract
- Multiply
- Divide
- Absolute Value
- Move
- Negate
- Compare

In addition, the FPA supports conversions between single- and double-precision floating-point formats and fixed-point formats.

The FPA incorporates separate Add/Subtract, Multiply, and Divide units, each capable of independent and concurrent operation. Thus, to achieve very high performance, floating point divides can be overlapped with floating point multiplies and floating point additions. These floating point operations occur independently of the actions of the CPU, allowing further overlap of integer and floating point operations. Figure 9 illustrates an example of the types of overlap permissible.

Exceptions

The FPA supports all five IEEE standard exceptions:

- Invalid Operation
- Inexact Operation
- Division by Zero
- Overflow
- Underflow

The FPA also suppoerts the optional, Unimplemented Operation exception that allows unimplemented instructions to trap to software emulation routines.

The FPA provides precise exception capability to the CPU; that is, the execution of a floating point operation which generates an exception causes that exception to occur at the CPU instruction which caused the operation. This precise exception capability is a requirement in applications and languages which provide a mechanism for local software exception handlers within software modules.

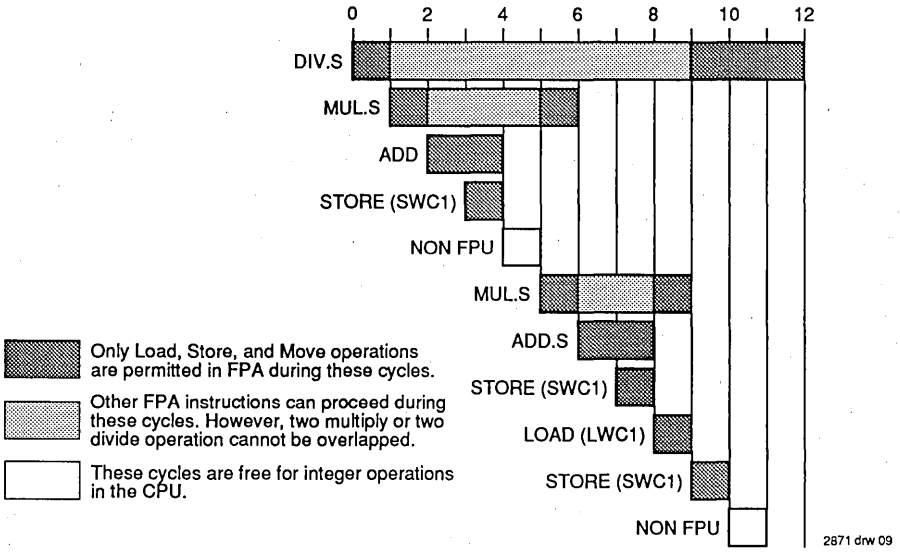


Figure 9. Examples of Overlapping Floating Point Operation

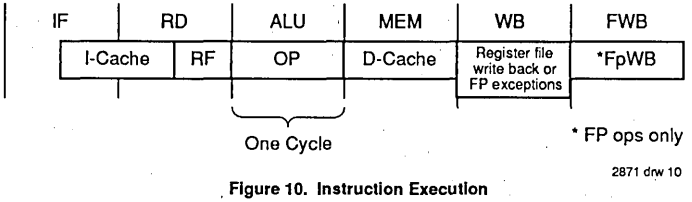


Figure 10. Instruction Execution

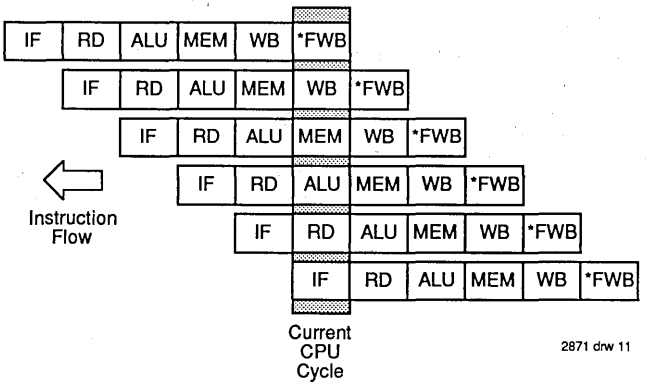


Figure 11. IDT79R3500A Execution Sequence

IDT79R3500 PIPELINE ARCHITECTURE

The execution of a single IDT79R3500A integer instruction consists of five pipe stages while floating point instruction takes six pipe stages. They are:

1) IF—Instruction fetch. The processor calculates the instruction address required to read from the I cache.
2) RD—The instruction is present on the data bus during phase one of this pipe stage. Instruction decode occurs during phase two. Operands are read from the registers if required.
3 ALU—Perform the required operation on instruction operands. If this is a FPA instruction, instruction execution commences.
4) MEM—Access memory. If the instruction is a load or store, the data is presented or captured during phase 2 of this pipe stage.
5) WB—Write integer results back into register file. In FPA cycles this pipe stage is used for exceptions.
6) FWB—The FPA uses this stage to write back ALU results to its register file.

Each of these steps requires approximately one FPA cycle as shown in Figure 10. (parts of some operations spill over into another cycle while other operations require only 1/2 cycle.)

The CPU uses a five stage pipeline while while the FPA uses a 6 stage to achieve an instruction execution rate approaching one instruction per cycle. Thus, execution of six instructions at a time are overlapped as shown in Figure 11.

This pipeline operates efficiently because different CPU resources (address and data bus accesses, ALU operations, register accesses, and so on) are utilized on a non-interfering basis.

MEMORY SYSTEM HIERARCHY

The high performance capabilities of the IDT79R3500A processor demand system configurations incorporating techniques frequently employed in large, mainframe computers but seldom encountered in systems based on more traditional microprocessors.

A primary goal of systems employing RISC techniques is to minimize the average number of cycles each instruction requires for execution. Techniques to reduce cycles-per-instruction include a compact and uniform instruction set, a deep instruction pipeline (as described above), and utilization of optimizing compilers. Many of the advantages obtained from these techniques can, however, be negated by an inefficient memory system.

Figure 12 illustrates memory in a simple microprocessor system. In this system, the CPU outputs addresses to memory and reads instructions and data from memory or writes data to memory. The address space is completely undifferentiated: instructions, data, and I/O devices are all treated the same. In such a system, a primary limiting performance factor is memory bandwidth.

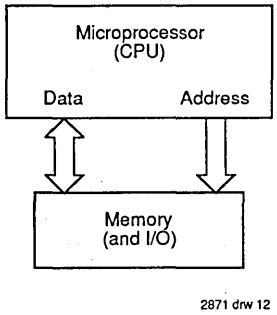


Figure 12. A Simple Microprocessor Memory System

Figure 13 illustrates a memory system that supports the significantly greater memory bandwidth required to take full advantage of the IDT79R3500A's performance capabilities. The key features of this system are:

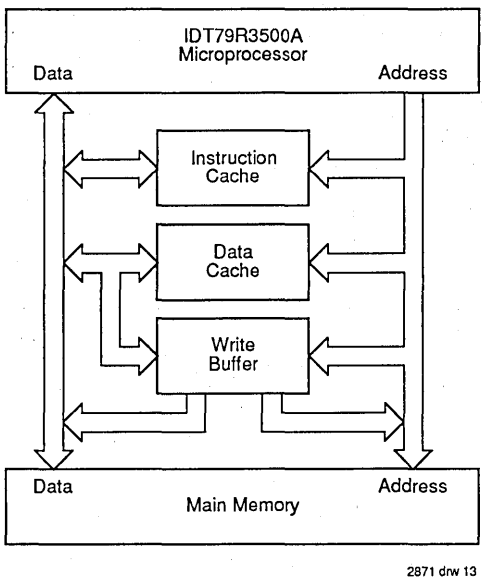


Figure 13. An IDT79R3500A System with a High-Performance Memory System

5

- **External Cache Memory**—Local, high-speed memory (called cache memory) is used to hold instructions and data that is repetitively accessed by the CPU (for example, within a program loop) and thus reduces the number of references that must be made to the slower-speed main memory. Some microprocessors provide a limited amount of cache memory on the CPU chip itself. The external caches supported by the IDT79R3500A can be much larger; while a small cache can improve performance of some programs, significant improvements for a wide range of programs require large caches.
- **Separate Caches for Data and Instructions**—Even with high-speed caches, memory speed can still be a limiting factor because of the fast cycle time of a high-performance microprocessor. The IDT79R3500A supports separate caches for instructions and data and alternates accesses of the two caches during each CPU cycle. Thus, the processor can obtain data and instructions at the cycle rate of the CPU using caches constructed with commercially available IDT static RAM devices.
  In order to maximize bandwidth in the cache while minimizing the requirement for SRAM access speed, the R3500A divides a single-processor clock cycle into two phases. During one phase, the address for the data cache access is presented while data previously addressed in the instruction cache is read; during the next phase, the data operation is completed while the instruction cache is being addressed. Thus, both caches are read in a single processor cycle using only one set of address and data pins.
- **Write Buffer**—in order to ensure data consistency, all data that is written to the data cache must also be written out to main memory. The cache write model used by the IDT79R3500A is that of a write-through cache; that is, all data written by the CPU is immediately written into the main memory. To relieve the CPU of this responsibility (and the inherent performance burden) the IDT79R3500A supports an interface to a write buffer. The IDT79R3020 Write Buffer captures data (and associated addresses) output by the CPU and ensures that the data is passed on to main memory.

IDT79R3500A Processor Subsystem Interfaces

Figure 14 illustrates the three subsystem interfaces provided by the IDT79R3500A processor:

- **Cache control** interface (on-chip) for separate data and instruction caches permits implementation of off-chip caches using standard IDT SRAM devices. The 79R3500A directly controls the cache memory with a minimum of external components. Both the instruction and data cache can vary from 0 to 256K Bytes (64K entries). The 79R3500A also includes the TAG control logic which determines whether or not the entry read from the cache is the desired data. The 79R3500A cache controller implements a direct mapped cache for high net performance (bandwidth). It has the ability to refill multiple words when a cache miss occurs, thus reducing the effective miss rate to less than 2% for large caches. When a cache miss occurs, the 79R3500A can support refilling the cache in 1, 4, 8, 16, or 32 word blocks to minimize the effective penalty of having to access main memory. The 79R3500A also incorporates the ability to perform instruction streaming; while the cache is refilling, the processor can resume execution once the missed word is obtained from main memory. In this way, the processor can continue to execute concurrently with the cache block refill.
- **Memory controller** interface for system (main) memory. This interface also includes the logic and signals to allow operation with a write buffer to further improve memory bandwidth. In addition to the standard full word access, the memory controller supports the ability to write bytes and half-words by using partial word operations. The memory controller also supports the ability to retry memory accesses if, for example, the data returned from memory is invalid and a bus error needs to be signalled.
- **Coprocessor Interface**—The IDT79R3500 features a set of on board tightly coupled coprocessors. Coprocessor 0 is defined to be the system control coprocessor and Coprocessor 1 is the Floating Point Accelerator. They have direct access to the internal data bus which allows them direct load and store of data in the same fashion as accessing the CPU registers. This relieves the typical bottleneck of having to load data into the CPU register set and then passing that data off to the co-processors.
  In applications where the FPA was off chip, as in using the IDT79R3010A several control pins were used for communications with the CPU and a Phase Lock Loop was located on the IDT79R3010A to synchronize the two together. As they are now integrated into a single chip, these are no longer needed. The FpCond output, which is used in coprocessor branch instructions, is now internally tied to the CpCond(1) input of the CPU leaving the external CpCond(1) pin available for another function. This signal is selectable to either output the FpBusy or the FPInt. For applications where FPInt was connected to any one of the six CPU HW interrupt inputs, that can also be internally routed-the default being Int(3), as recommended by the MIPS architecture. If FPInt is internally routed, the external interrupt input corresponding to the FP interrupt is ignored. Internal routing of these selections are made via the reset vector.
  The internal CPBusy input, which is used to stall the CPU if the coprocessor needs to hold off subsequent operations, has two sources-FPBusy and the external CpBusy pin which are logically ORed together. Further, Run and Exception of both the FPA and CPU are internally tied and brought out with the external CPBusy input to accommodate off chip coprocessor 2 and 3. This external interface is available to support application specific functions.

MULTIPROCESSING SUPPORT

The IDT79R3500A supports multiprocessing applications in a simple but effective way. Multiprocessing applications require cache coherency across the multiple processors. The IDT79R3500A offers two signals to support cache coherency: the first, MPStall, stalls the processor within two cycles of being received and keeps it from accessing the cache. This allows an external agent to snoop into the processor data cache. The second signal, MPInvalidate, causes the processor to write data on the data cache bus which indicates the externally addressed cache entry is invalid. Thus, a subsequent access to that location would result in a cache miss, and the data would be obtained from main memory.

The two MP signals would be generated by a external logic which utilizes a secondary cache to perform bus snooping functions. The 79R3500A does not impose an architecture for this secondary cache, but rather is flexible enough to support a variety of application specific architectures and still maintain cache coherency. Further, there is no impact on designs which do not require this feature. The 79R3500A further allows the use of cache RAMs with internal address latches in multiprocessor systems.

ADVANCED FEATURES

The IDT79R3500A offers a number of additional features such as the ability to swap the instruction and data caches, facilitating diagnostics and cache flushing. Another feature isolates the, caches, which forces cache hits to occur regardless of the contents of the tag fields. The IDT79R3500A allows the processor to execute user tasks of the opposite byte ordering (endianness) of the operating system, has double the integer multiply/divide performance of R3000 and R2000, has a programmable Tag width bus, and further allows parity checking to be disabled. More details on these features can be found in the IDT79R3500 Family Hardware User's Manual.

Further features of the IDT79R3500A are configured during the last four cycles prior to the negation of the RESET input. These functions include the ability to select cache sizes and cache refill block sizes; the ability to utilize the multiprocessor interface; whether or not instruction streaming is enabled; whether byte ordering follows "Big-Endian" or "Little-Endian" protocols, etc. Table 3 shows the configuration options selected at Reset. These are further discussed in the "Hardware User's Manual".

BACKWARD COMPATIBILITY

The primary goal of the 79R3500A is the ability to replace the R3000 and R3010 with a single chip solution. This can be done either the R3000/R3010 or the R3000A/R3010A as well. The pinout of the IDT79R3500A has been selected to ensure this compatibility, with new functions mapped onto previously used pins. The instruction set is compatible with that of the R2000 at the binary level. As a result, code written for the older processor can be executed.

In most R3000A applications, the IDT79R3500 can be placed in the socket with no modification to initialization settings. Additionally, the IDT79R3500 can be used in systems that did not include the R3010 in the original design. Further application assistance on these topics are available from IDT.

PACKAGE THERMAL SPECIFICATIONS

The IDT79R3500A utilizes special packaging techniques to improve both the thermal and electrical characteristics of the microprocessor.

In order to improve the electrical characteristics of the device, the package is constructed using multiple signal planes, including individual power planes and ground planes to reduce noise associated with high-frequency TTL parts. In addition, the 175-pin PGA package utilizes extra power and ground pins to reduce the inductance from the internal power planes to the power planes of the PC Board.

In order to improve the electrical characteristics of the microprocessor, the device is housed using cavity down packaging. In addition, these packages incorporate a copper-tungsten thermal slug designed to efficiently transfer heat from the die to the case of the package, and thus effectively lower the thermal resistance of the package. The use of an additional external heat sink affixed to the package thermal slug further decreases the effective thermal resistance of the package.

The case temperature may be measured in any environment to determine whether the device is within the specified operating range. The case temperature should be measured at the center of the top surface opposite the package cavity (the package cavity is the side where the package lid is mounted).

The equivalent allowable ambient temperature, T_A, can be calculated using the thermal resistance from case to ambient ($\varnothing ca$) for the given package. The following equation relates ambient and case temperature:

$$T_A = T_c - P * \varnothing ca$$

where P is the maximum power consumption, calculated by using the maximum Icc from the DC Electrical Characteristics section.

Typical values for $\varnothing ca$ at various airflows are shown in table 2 for the various CPU packages.

	Airflow - (ft/min)					
	0	200	400	600	800	1000
$\varnothing ca$ (175-PGA, 144-PGA)	21	7	3	2	1	0.5
$\varnothing ca$ (172 Quad Flatpack)	23	9	4	3	2.5	1.5

2871 tbl 03

Table 2. R3500A Package Characteristics

Input	W Cycle	X Cycle	Y Cycle	Z Cycle
Int0	DBlkSize0	DBlkSize1	Extend Cache	Big Endian
Int1	IBlkSize0	IBlkSize1	MPAdrDisable	TriState
Int2	DispPar/RevEnd	IStream	IgnoreParity	NoCache
Int3	Reserved(1)	StorePartial	MultiProcessor	BusDriveOn
Int4	FPINT decode	FPINT decode	FPINT decode	FPINT onto CpCond
Int5	7RR3500 mode	TLB disable	Tag Mode 1	Tag Mode 0

2871 tbl 04

NOTES:
1. Reserved entries must be driven high.
2. These values must be driven stable throughout the enfire RESET period.

Table 3. R3500A Mode Selectable Features

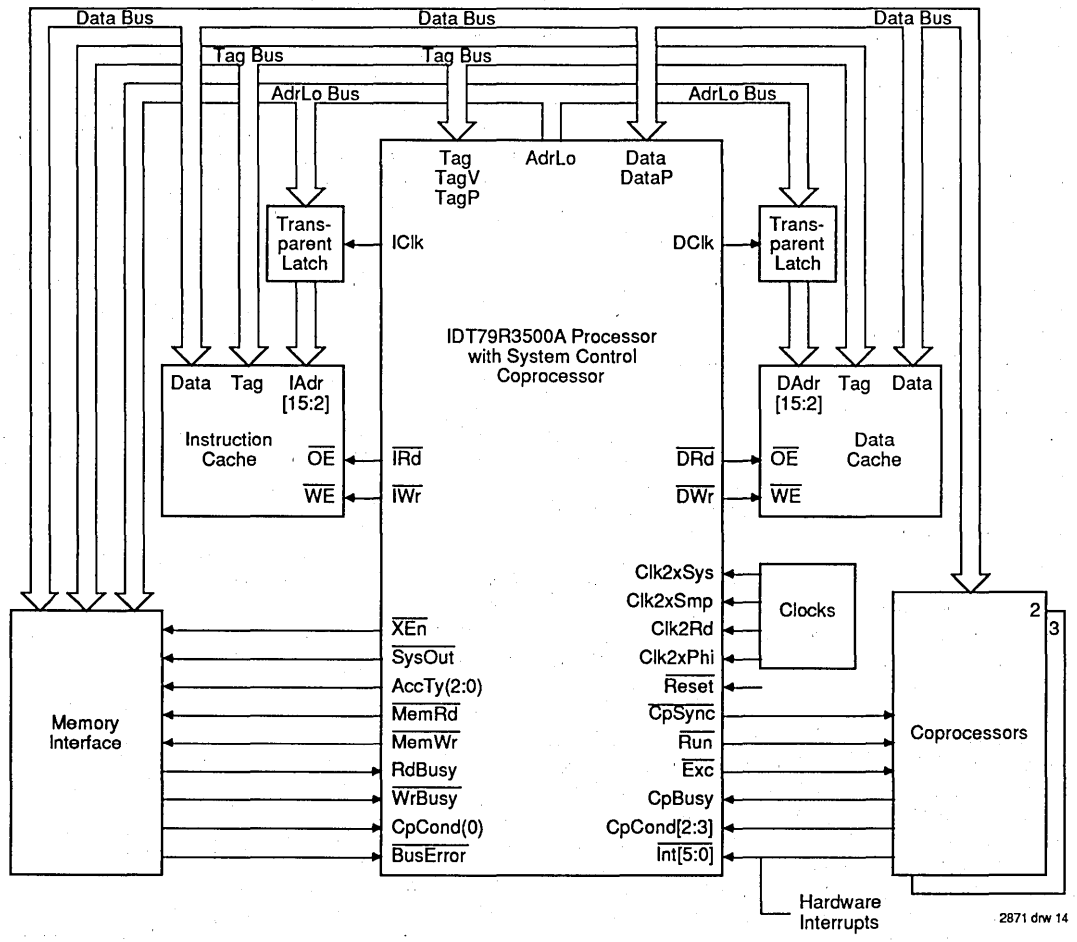
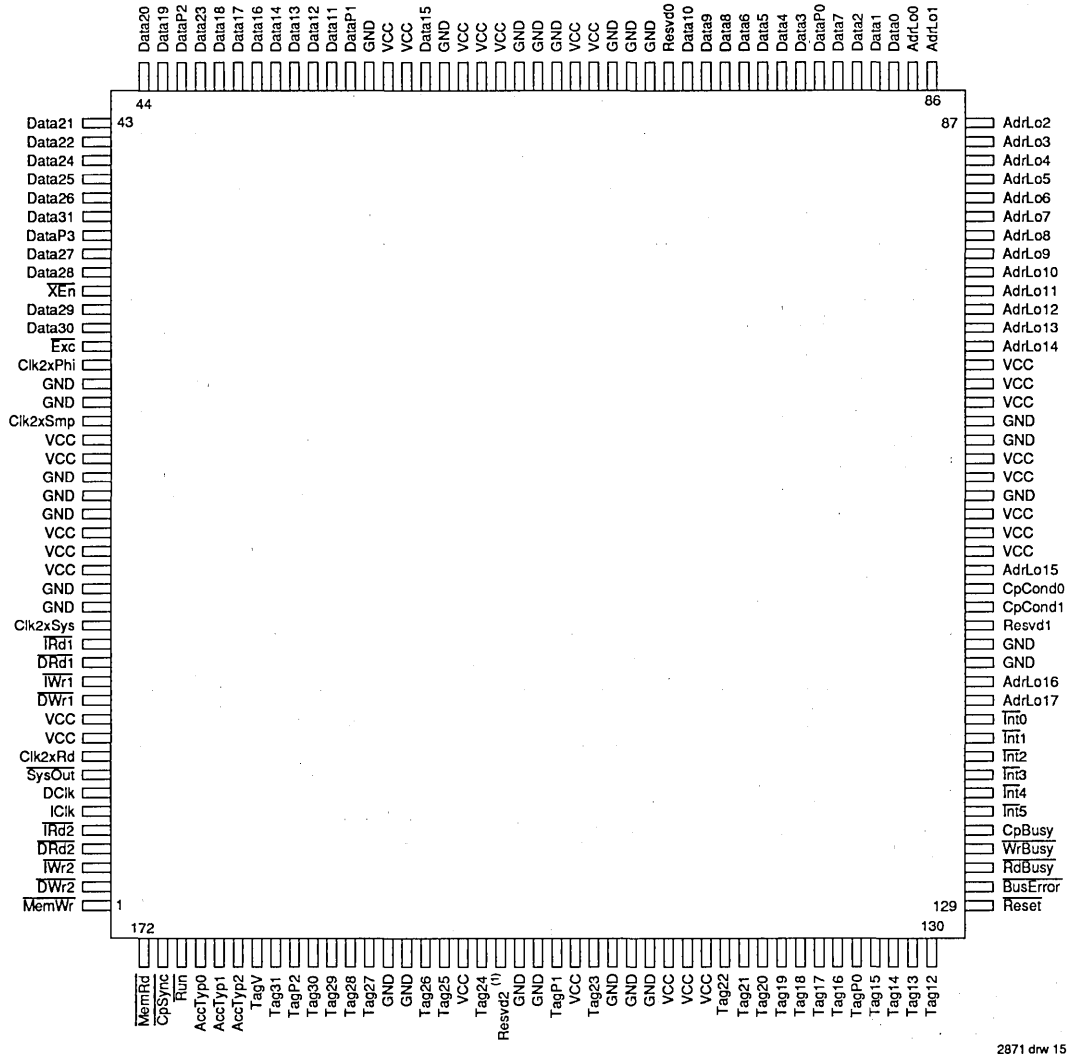


Figure 14. IDT79R3500A Subsystem Interfaces Example; 64 KB Caches

PIN CONFIGURATION



2871 drw 15

172-Pin Flatpack (Top View)

NOTES:
1. Reserved pins must be connected.
2. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN CONFIGURATION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	(No Pin)	AdrLo 6	AdrLo 10	AdrLo 11	VCC	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	$\overline{\text{Int(2)}}$	$\overline{\text{Int(5)}}$	Wr Busy	$\overline{\text{Reset}}$	VCC
B	AdrLo 3	$\overline{\text{DRd2}}$	AdrLo 7	AdrLo 9	AdrLo 12	$\overline{\text{IRd2}}$	AdrLo 13	CpCond 1	$\overline{\text{Int(1)}}$	$\overline{\text{Int(3)}}$	Cp Busy	$\overline{\text{Bus}}$ $\overline{\text{Error}}$	$\overline{\text{DWr2}}$	Tag12	Tag15
C	AdrLo 0	AdrLo 4	VCC	AdrLo 5	AdrLo 8	GND	GND	VCC	$\overline{\text{Int(0)}}$	$\overline{\text{Int(4)}}$	Rd Busy	GND	Tag13	TagP0	Tag18
D	Data 1	AdrLo 2	GND	GND	VCC	GND	VCC	GND	VCC	GND	VCC	GND	Tag14	Tag17	Tag19
E	DataP 0	Data 0	AdrLo 1	VCC								VCC	Tag16	Tag20	VCC
F	VCC	Data 7	Data 2	GND								GND	GND	Tag21	Tag23
G	Data 4	Data 3	GND	VCC								VCC	GND	Tag22	TagP1
H	Data 6	Data 5	Data 8	GND								GND	VCC	Tag25	Tag24
J	Data 10	DataP 1	Data 9	VCC								VCC	Tag28	Tag29	Tag26
K	Data 15	Data 11	GND	GND								GND	GND	TagP2	Tag27
L	VCC	Data 12	Data 17	VCC								VCC	Acc Typ2	Tag31	Tag30
M	Data 13	Data 16	DataP 2	GND	VCC	GND	VCC	GND	VCC	GND	VCC	GND	GND	Acc Typ1	VCC
N	Data 14	Data 18	Data 19	GND	Data 24	DataP 3	VCC	VCC	GND	GND	$\overline{\text{DRd1}}$	$\overline{\text{Mem}}$ $\overline{\text{Wr}}$	$\overline{\text{Mem}}$ $\overline{\text{Rd}}$	$\overline{\text{Run}}$	TagV
P	Data 23	Data 20	$\overline{\text{IWr2}}$	Data 22	Data 26	Data 27	$\overline{\text{XEn}}$	Data 30	Clk2x Sys	Clk2x Rd	DClk	$\overline{\text{IRd1}}$	$\overline{\text{IWr1}}$	$\overline{\text{Cp}}$ $\overline{\text{Sync}}$	Acc Typ0
Q	VCC	Data 21	Data 25	Data 31	Data 28	GND	Data 29	$\overline{\text{Excep-}}$ $\overline{\text{tion}}$	Clk2x Phi	Clk2x Smp	$\overline{\text{SysOut}}$	VCC	IClk	$\overline{\text{DWr1}}$	VCC

2871 drw 16

175-Pin PGA (Top View)

NOTE:
1. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN CONFIGURATION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	VCC	AdrLo 6	AdrLo 10	AdrLo 11	VCC	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	$\overline{\text{Int(2)}}$	$\overline{\text{Int(5)}}$	Wr Busy	$\overline{\text{Reset}}$	VCC
B	AdrLo 3	$\overline{\text{DRd2}}$	AdrLo 7	AdrLo 9	AdrLo 12	$\overline{\text{IRd2}}$	AdrLo 13	CpCond 1	$\overline{\text{Int(1)}}$	$\overline{\text{Int(3)}}$	Cp Busy	$\overline{\text{Bus}}$ $\overline{\text{Error}}$	$\overline{\text{DWr2}}$	Tag12	Tag15
C	AdrLo 0	AdrLo 4	VCC	AdrLo 5	AdrLo 8	GND	GND	VCC	$\overline{\text{Int(0)}}$	$\overline{\text{Int(4)}}$	Rd Busy	GND	Tag13	TagP0	Tag18
D	Data 1	AdrLo 2	GND	GND									Tag14	Tag17	Tag19
E	DataP 0	Data 0	AdrLo 1										Tag16	Tag20	VCC
F	VCC	Data 7	Data 2										GND	Tag21	Tag23
G	Data 4	Data 3	GND										GND	Tag22	TagP1
H	Data 6	Data 5	Data 8										VCC	Tag25	Tag24
J	Data 10	DataP 1	Data 9										Tag28	Tag29	Tag26
K	Data 15	Data 11	GND										GND	TagP2	Tag27
L	VCC	Data 12	Data 17										Acc Typ2	Tag31	Tag30
M	Data 13	Data 16	DataP 2										GND	Acc Typ1	VCC
N	Data 14	Data 18	Data 19	GND	Data 24	DataP 3	VCC	VCC	GND	GND	$\overline{\text{DRd1}}$	$\overline{\text{Mem}}$ $\overline{\text{Wr}}$	$\overline{\text{Mem}}$ $\overline{\text{Rd}}$	$\overline{\text{Run}}$	TagV
P	Data 23	Data 20	$\overline{\text{IWr2}}$	Data 22	Data 26	Data 27	$\overline{\text{XEn}}$	Data 30	Clk2x Sys	Clk2x Rd	DClk	$\overline{\text{IRd1}}$	$\overline{\text{IWr1}}$	$\overline{\text{Cp}}$ $\overline{\text{Sync}}$	Acc Typ0
Q	VCC	Data 21	Data 25	Data 31	Data 28	GND	Data 29	$\overline{\text{Excep-}}$ $\overline{\text{tion}}$	Clk2x Phi	Clk2x Smp	$\overline{\text{SysOut}}$	VCC	IClk	$\overline{\text{DWr1}}$	VCC

2871 drw 17

144-Pin PGA (Top View)

NOTE:
1. AdrLo 16 and 17 are multifunction pins which are controlled by mode select programming on interrupt pins at reset time
   AdrLo 16: MP Invalidate, CpCond (2).
   AdrLo 17: MP Stall, CpCond (3).

PIN DESCRIPTIONS

Pin Name	I/O	Description
Data (0-31)	I/O	A 32-bit bus used for all instruction and data transmission among the processor, caches, memory interface, and coprocessors.
DataP (0-3)	I/O	A 4-bit bus containing even parity over the data bus.
Tag (12-31)	I/O	A 20-bit bus used for transferring cache tags and high addresses between the processor, caches, and memory interface.
TagV	I/O	The tag validity indicator.
Tag P (0-2)	I/O	A 3-bit bus containing even parity over the concatenation of TagV and Tag.
AdrLo (0-17)	O	An 18-bit bus containing byte addresses used for transferring low addresses from the processor to the caches and memory interface. (AdrLo 16: CpCond (2), AdrLo 17: CpCond (3) set by reset initialization).
$\overline{\text{IRd1}}$	O	Read enable for the instruction cache.
$\overline{\text{IWr1}}$	O	Write enable for the instructon cache.
$\overline{\text{IRd2}}$	O	An identical copy of $\overline{\text{IRd1}}$ used to split the load.
$\overline{\text{IWr2}}$	O	An identical copy of $\overline{\text{IWr1}}$ used to split the load.
IClk	O	The instruction cache address latch clock. This clock runs continuously.
$\overline{\text{DRd1}}$	O	The read enable for the data cache.
$\overline{\text{DWr1}}$	O	The write enable for the data cache.
$\overline{\text{DRd2}}$	O	An identical copy of $\overline{\text{DRd1}}$ used to split the load.
$\overline{\text{DWr2}}$	O	An identical copy of $\overline{\text{DWr1}}$ used to split the load.
DClk	O	The data cache address latch clock. This clock runs continuously.
$\overline{\text{XEn}}$	O	The read enable for the Read Buffer.
AccTyp(0-2)	O	A 3-bit bus used to indicate the size of data being transferred on the data bus, whether or not a data transfer is occurring, and the purpose of the transfer.
$\overline{\text{MemWr}}$	O	Signals the occurrence of a main memory write.
$\overline{\text{MemRd}}$	O	Signals the occurrence of a main memory read.
$\overline{\text{BusError}}$	I	Signals the occurrence of a bus error during a main memory read or write.
$\overline{\text{Run}}$	O	Indicates whether the processor is in the RUN or STALL state. In the discrete design, the R3000 $\overline{\text{Run}}$ output is tied directly to the R3010 $\overline{\text{Run}}$ input. In the 79R3500, this is done internally, but the $\overline{\text{Run}}$ signal is also brought out for application specific coprocessors.
$\overline{\text{Exception}}$	O	Indicates that the instruction that is about to commit to a state change should be aborted; also indicates other exception related information. In the discrete design, the R3000 $\overline{\text{Exception}}$ output is tied to the R3010 $\overline{\text{Exception}}$ input. In the 79R3500 this is done internally, but the $\overline{\text{Exception}}$ signal is also brought out for application specific coprocessors.
$\overline{\text{CpSync}}$	O	A clock which is identical to $\overline{\text{SysOut}}$ and used by external coprocessors for timing synchronization with the 79R3500. In the discrete design, $\overline{\text{CpSync}}$ output from the R3000 is tied to the R3010 $\overline{\text{FPSync}}$ input. In the 79R3500, this is done internally, but the $\overline{\text{CpSync}}$ signal is also brought out for application specific coprocessors.
RdBusy	I	The main memory read stall termination signal. In most system designs RdBusy is normally asserted and is deasserted only to indicate the successful completion of a memory read. RdBusy is sampled by the processor only during memory read stalls.
$\overline{\text{WrBusy}}$	I	The main memory write stall initiation/termination signal.
$\overline{\text{CpBusy}}$	I	Input used to indicate that the requested coprocessor resource is unavailable, or used to preserve the precise exception model. In the descrete design, $\overline{\text{CpBusy}}$ is driven directly by the R3010 $\overline{\text{FpBusy}}$ output. In the 79R3500 the $\overline{\text{CpBusy}}$ input of the CPU is the logical OR of both the internal FPA $\overline{\text{FpBusy}}$ and the external $\overline{\text{CpBusy}}$ pin. This input is provided for external application specific coprocessors. An internal pull down resistor is provided if this input is left open.
CpCond(1)	I	Signal used by the branch on Coprocessor 1 true/false instruction. In discrete systems using a R3010 FPA, this is normally tied to the FpCond output. In the 79R3500, the internal $\overline{\text{FpCond}}$ is directly tied to the internal CpCond(1) input leaving this pin available for other functions. This pin defaults to output the FpBusy internal signal or, (via the Reset vectors), output the FPInt—in the latter case, external hardware must route this signal to the appropriate Int pin.
CpCond (0,2-3)	I	Conditional branch status from coprocessors to the processor. Function is provided on AdrLo 16/17 pins and is selected at reset time.
MPStall	I	Multiprocessing Stall. Signals to the processor that it should stall accesses to the caches in a multiprocessing environment. This is physically the same pin as CpCond3; its use is determined at RESET initialization.
MPInvalidate	I	Multiprocessing Invalidate. Signals to the processor that it should issue invalidate data on the cache data bus. The address to be invalidated is externally provided. This is the same pin as CpCond2; its use is determined at RESET initialization.
$\overline{\text{Int (0-5)}}$	I	A 6-bit bus used by the memory interface and coprocessors to signal maskable interrupts to the 79R3500. This bus is also used at reset time to select among the mode-selectable features of the 79R3500. The FPA $\overline{\text{FPInt}}$ output signal is typically connected to one of these interrupt lines; the choice is programmable through the reset vectors with the default being Int(3).

PIN DESCRIPTIONS (Continued)

Pin Name	I/O	Description
Clk2xSys	I	The master double frequency input clock used for generating SysOut.
Clk2xSmp	I	A double frequency clock input used to determine the sample point for data coming into the processor and coprocessors.
Clk2xRd	I	A double frequency clock input used to determine the enable time of the cache RAMs.
Clk2xPhi	I	A double frequency clock input used to determine the position of the internal phases, phase1 and phase2.
Reset	I	Synchronous initialization input used to force execution starting from the reset memory address. Reset must be deasseted synchronously but asserted asynchronously. The deassertion of Reset must be synchronized by the leading edge of SysOut.

2871 tbl 05

ABSOLUTE MAXIMUM RATINGS(1, 3)

Symbol	Rating	Commercial	Military	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70(4) (Ambient) 0 to +90(5) (Case)	–55 to +125 (Case)	°C
TBIAS	Case Temperature Under Bias	–55 to +125(4) 0 to +90(5)	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
IIN	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

2871 tbl 06

NOTE:
1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. VIN minimum = -3.0V for pulse width less than 15ns.
   VIN should not exceed VCC +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.
4. 16-33 MHz only.
5. 37-40 MHz only.

AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V

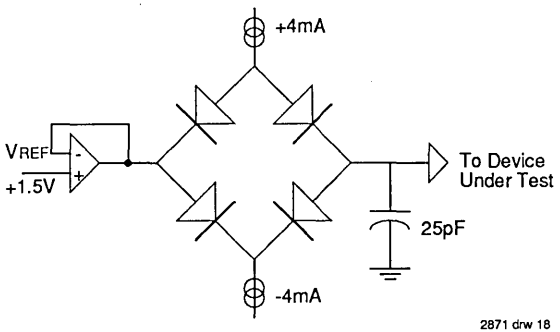
2871 tbl 07

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	Vcc
Military 16-33 MHZ	-55°C to +125°C (Case)	0V	5.0 ±10%
Commercial 16-33 MHz	0°C to +70°C (Ambient)	0V	5.0 ±5%
Commercial 37-40 MHz	0°C to +90°C (Case)	0V	5.0 ±5%

2871 tbl 08

OUTPUT LOADING FOR AC TESTING



2871 drw 18

DC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

			79R3500A				79R3500AE				
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
Symbol	Parameter	Test Conditions	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
C_{IN}	Input Capacitance(6)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	450	—	550	—	650	—	750	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	-100	100	-100	100	µA

2871 tbl 09

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag busses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

DC ELECTRICAL CHARACTERISTICS— MILITARY TEMPERATURE RANGE (TC = -55°C to +125°C, VCC = +5.0V ±10%)

			79R3500A				79R3500AE				
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
Symbol	Parameter	Test Conditions	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Unit
VOH	Output HIGH Voltage	VCC = Min., IOH = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
VOL	Output LOW Voltage	VCC = Min., IOL = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
VOHC	Output HIGH Voltage(7)	VCC = Min., IOH = -4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
VOHT	Output HIGH Voltage(4,6)	VCC = Min., IOH = -8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
VOLT	Output LOW Voltage(4,6)	VCC = Min., IOL = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
VIH	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
VIL	Input LOW Voltage(1)		—	0.8	—	0.8	—	0.8	—	0.8	V
VIHS	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
VILS	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
CIN	Input Capacitance(6)		—	10	—	10	—	10	—	10	pF
COUT	Output Capacitance(6)		—	10	—	10	—	10	—	10	pF
ICC	Operating Current	VCC = 5V, TA = 70°C	—	500	—	600	—	650	—	750	mA
IIH	Input HIGH Leakage(3)	VIH = VCC	—	100	—	100	—	100	—	100	µA
IIL	Input LOW Leakage(3)	VIL = GND	-100	—	-100	—	-100	—	-100	—	µA
IOZ	Output Tri-state Leakage	VOH = VCC, VOL = GND	-100	100	-100	100	-100	100	-100	100	µA

2871 tbl 10

5

NOTES:

1. VIL Min. = -3.0V for pulse width less than 15ns. VIL should not fall below -0.5 Volts for larger periods.
2. VIHS and VILS apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. VOHT and VOLT apply to the bidirectional data and tag busses only. Note that VIH and VIL also apply to these signals. VOHT and VOLT are provided to give the designer further information about these specific signals.
5. VIH should not be held above VCC + 0.5 volts.
6. Guaranteed by design.
7. VOHC applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

DC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE (T_C = 0°C to +90°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3500AE				Unit
			37.0MHz		40.0MHz		
			Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	V
C_{IN}	Input Capacitance(6)		—	10	—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	825	—	850	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	-100	100	-100	100	µA

2871 tbl 11

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag busses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3500A 16.67MHz Min.	79R3500A 16.67MHz Max.	79R3500A 20.0MHz Min.	79R3500A 20.0MHz Max.	79R3500AE 25.0MHz Min.	79R3500AE 25.0MHz Max.	79R3500AE 33.33MHz Min.	79R3500AE 33.33MHz Max.	Unit
Clock											
TckHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		9	tcyc/4	7	tcyc/4	5	tcyc/4	3.5	tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	2	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type (2)	Load= 25pF	—	17	—	14	—	12	—	8.5	ns
TMWr	Memory Write	Load= 25pF	—	27	—	23	—	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
TAval	Address Valid	Load= 25pF	—	2	—	2	—	1.5	—	1	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		9	—	8	—	6	—	4.5	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	1	18	1	13.5	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	27	—	23	—	18	—	10	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSExc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	128	—	128	—	Tcyc
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	2	0.5	1	0.5	1	0	1	ns/25pF

2871 tbl 12

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33.33MHz; clock transition time < 5ns for other speeds.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— MILITARY TEMPERATURE RANGE (Tc = -55°C to +125°C, Vcc = +5.0V ±10%)

Symbol	Parameter	Test Conditions	79R3500A				79R3500AE				Unit
			16.67MHz		20.0MHz		25.0MHz		33.33MHz		
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Clock											
TckHigh	Input Clock High[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckLow	Input Clock Low[2]	Note 7	12.5	—	10	—	8	—	6	—	ns
TckP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		9	tcyc/4	7	tcyc/4	5	tcyc/4	3.5	tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	-2	—	-2	—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	3	—	3	—	3	—	2	ns
TWrDly	Write Delay	Load= 25pF	—	5	—	4	—	3	—	2	ns
TDS	Data Set-up		9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7	—	6	—	5	—	3.5	ns
TAT2	Access Type (2)	Load= 25pF	—	17	—	14	—	12	—	8.5	ns
TMWr	Memory Write	Load= 25pF	—	27	—	23	—	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7	—	7	—	5	—	3.5	ns
TAval	Address Valid	Load= 25pF	—	2	—	2	—	1.5	—	1	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		9	—	8	—	6	—	4.5	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	-2.5	—	-2.5	—	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1	27	1	23	—	18	—	13.5	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	27	—	23	—	18	—	10	ns
TStl	Run Terminate	Load= 25pF	3	17	3	15	3	10	2	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	Load= 25pF	3	27	3	23	3	18	2	9.5	ns
TSExc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
Reset Initialization											
TRST	Reset Pulse Width		6	—	6	—	6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	128	—	128	—	Tcyc
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	2	0.5	1	0.5	1	0	1	ns/25pF

2871 tbl 13

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns for 33.33MHz; clock transition time < 5ns for other speeds.

AC ELECTRICAL CHARACTERISTICS[1,2,3]— COMMERCIAL TEMPERATURE RANGE (Tc = 0°C to +90°C, Vcc = +5.0V ±5%)

Symbol	Parameter	Test Conditions	79R3500AE 37.0MHz Min.	79R3500AE 37.0MHz Max.	79R3500AE 40.0MHz Min.	79R3500AE 40.0MHz Max.	Unit
Clock							
TCkHigh	Input Clock High[2]	Note 7	5.5	—	5	—	ns
TCkLow	Input Clock Low[2]	Note 7	5.5	—	5	—	ns
TCkP	Input Clock Period[2]		13.5	500	12.5	500	ns
	Clk2xSys to Clk2XSmp[6]		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd[6]		0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi[6]		3.5	tcyc/4	3	tcyc/4	ns
Run Operation							
TDEn	Data Enable[3]		—	-1.5	—	-1.5	ns
TDDis	Data Disable[3]		—	-0.5	—	-0.5	ns
TDVal	Data Valid	Load= 25pF	—	2	—	1.5	ns
TWrDly	Write Delay	Load= 25pF	—	2	—	2	ns
TDS	Data Set-up		4.5	—	4	—	ns
TDH	Data Hold[3]		-2.5	—	-2.5	—	ns
TCBS	CpBusy Set-up		6	—	6	—	ns
TCBH	CpBusy Hold		-2.5	—	-2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	3.5	—	3	ns
TAT2	Access Type (2)	Load= 25pF	—	8.5	—	7.5	ns
TMWr	Memory Write	Load= 25pF	—	9.5	—	9	ns
TExc	Exception	Load= 25pF	—	3.5	—	3	ns
TAval	Address Valid	Load= 25pF	—	1	—	0.5	ns
TIntS	$\overline{\text{Int(n)}}$ Set-up		4.5	—	4	—	ns
TIntH	$\overline{\text{Int(n)}}$ Hold		-2.5	—	-2.5	—	ns
Stall Operation							
TSAVal	Address Valid	Load= 25pF	—	15	—	12.5	ns
TSAcTy	Access Type	Load= 25pF	—	13.5	—	9	ns
TMRdi	Memory Read Initiate	Load= 25pF	—	13.5	—	9	ns
TMRdt	Memory Read Terminate	Load= 25pF	—	10	—	9	ns
TStl	Run Terminate	Load= 25pF	2	6.5	2	6	ns
TRun	Run Initiate	Load= 25pF	—	3	—	3	ns
TSMWr	Memory Write	Load= 25pF	2	9.5	2	9	ns
TSExc	Exception Valid	Load= 25pF	—	6.5	—	6	ns
Reset Initialization							
TRST	Reset Pulse Width		6	—	6	—	Tcyc
TrstPLL	Reset timing, Phase-lock on[4,5]		3000	—	3000	—	Tcyc
Trstcp	Reset timing, Phase-lock off[4,5]		128	—	128	—	Tcyc
Capacitive Load Deration							
CLD	Load Derate[6]		0.5	1	0	1	ns/25pF

2871 tbl 14

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all four 2xClocks: Clk2xSys, Clk2xSmp, Clk2xRd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters apply when the 79R3010 Floating Point Coprocessor is connected to the CPU. With phase lock on, $\overline{\text{Reset}}$ must be asserted for the longer of 3000 clock cycles or 200 microseconds.
5. Tcyc is one CPU clock cycle (two cycles of a 2x clock).
6. With the exception of the $\overline{\text{Run}}$ signal, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Clock transition time < 2.5ns.

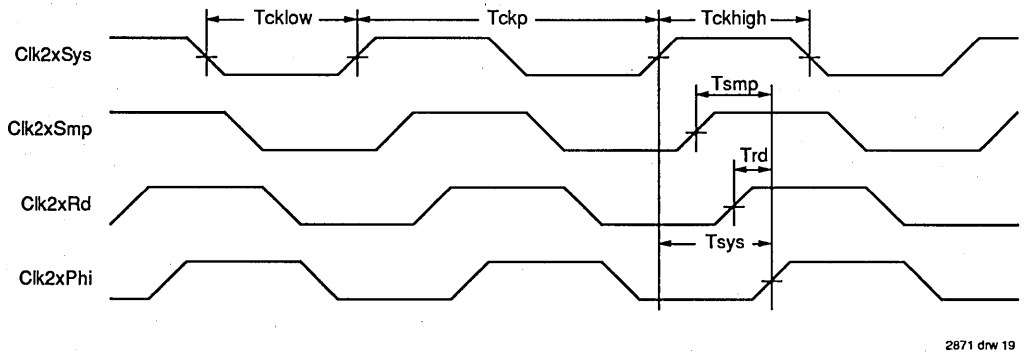


Figure 15. Input Clock Timing

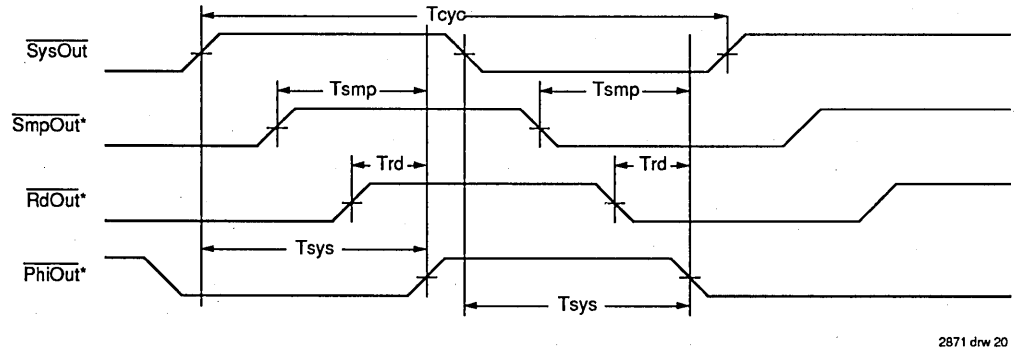


Figure 16. Processor Reference Clock Timing

* These signals are not actually output from the processor.
They are drawn to provide a reference for other timing diagrams.

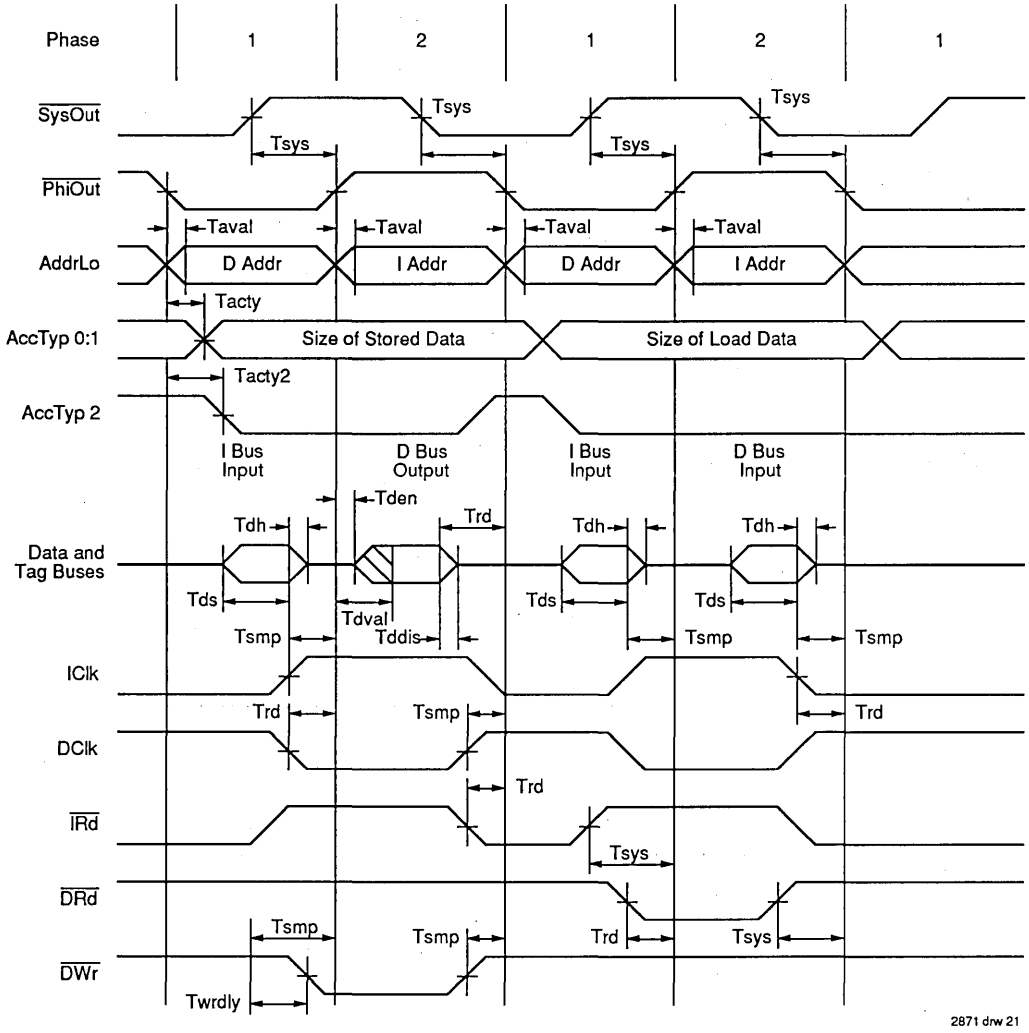


Figure 17. Synchronous Memory (Cache) Timing

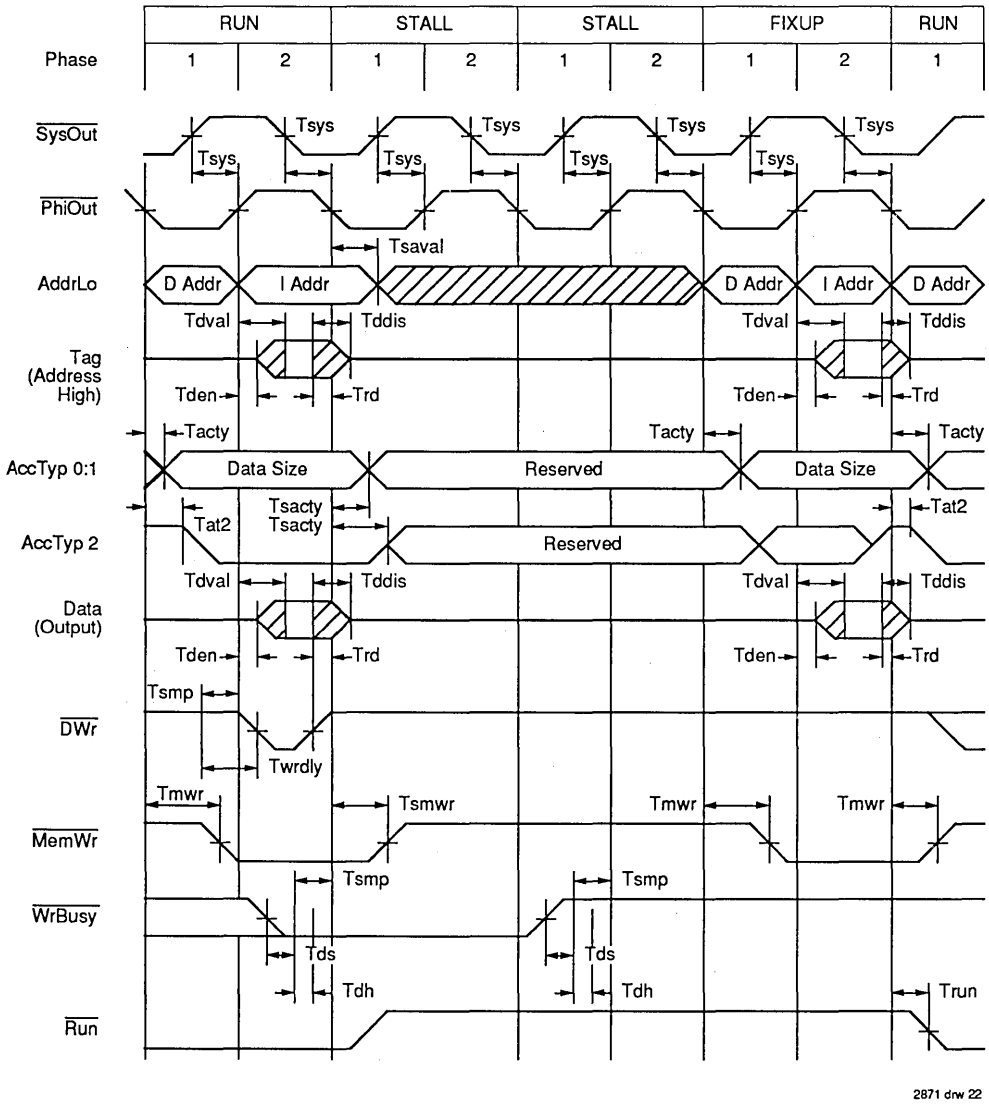


Figure 18. Memory Write Timing

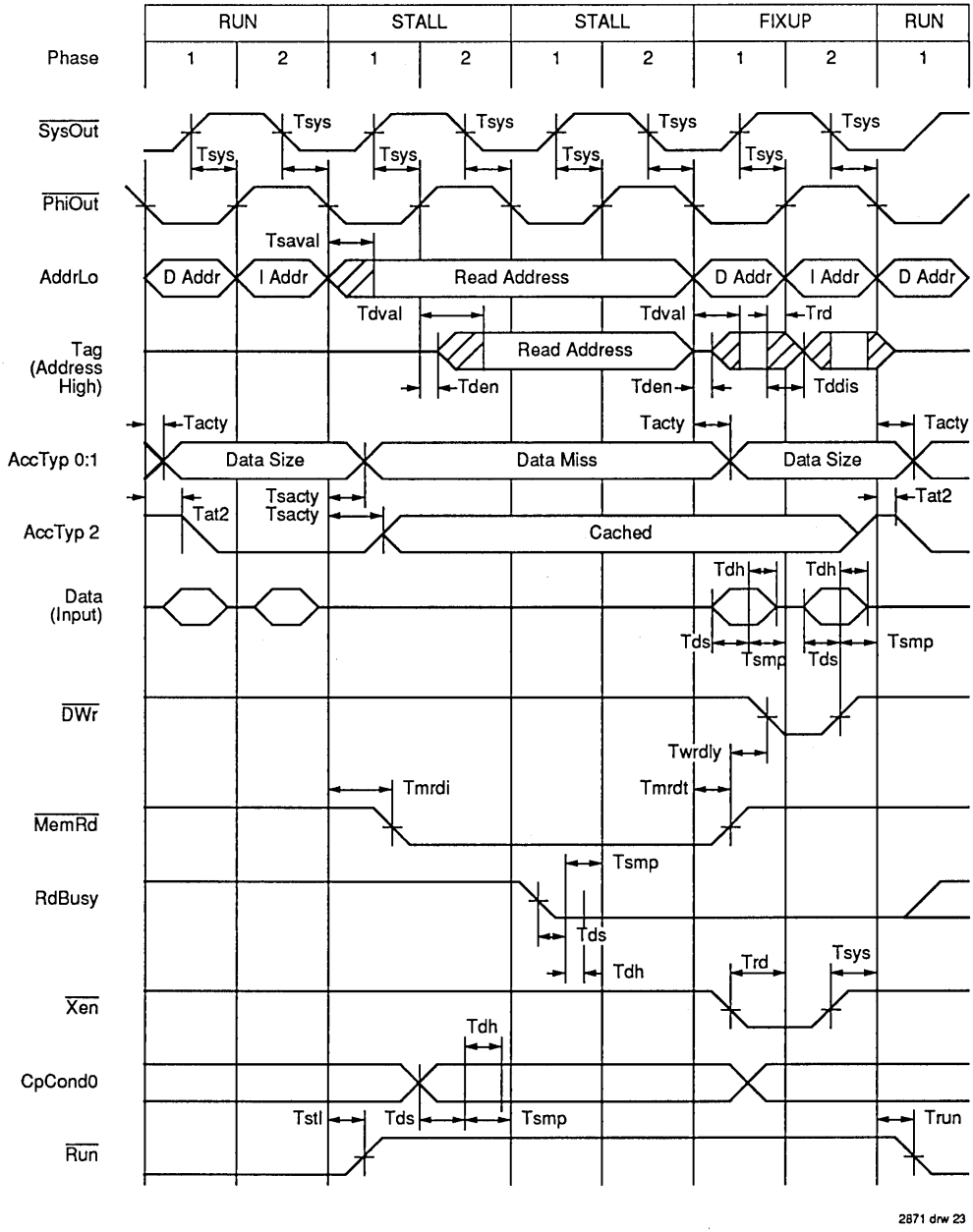


Figure 19. Memory Read Timing

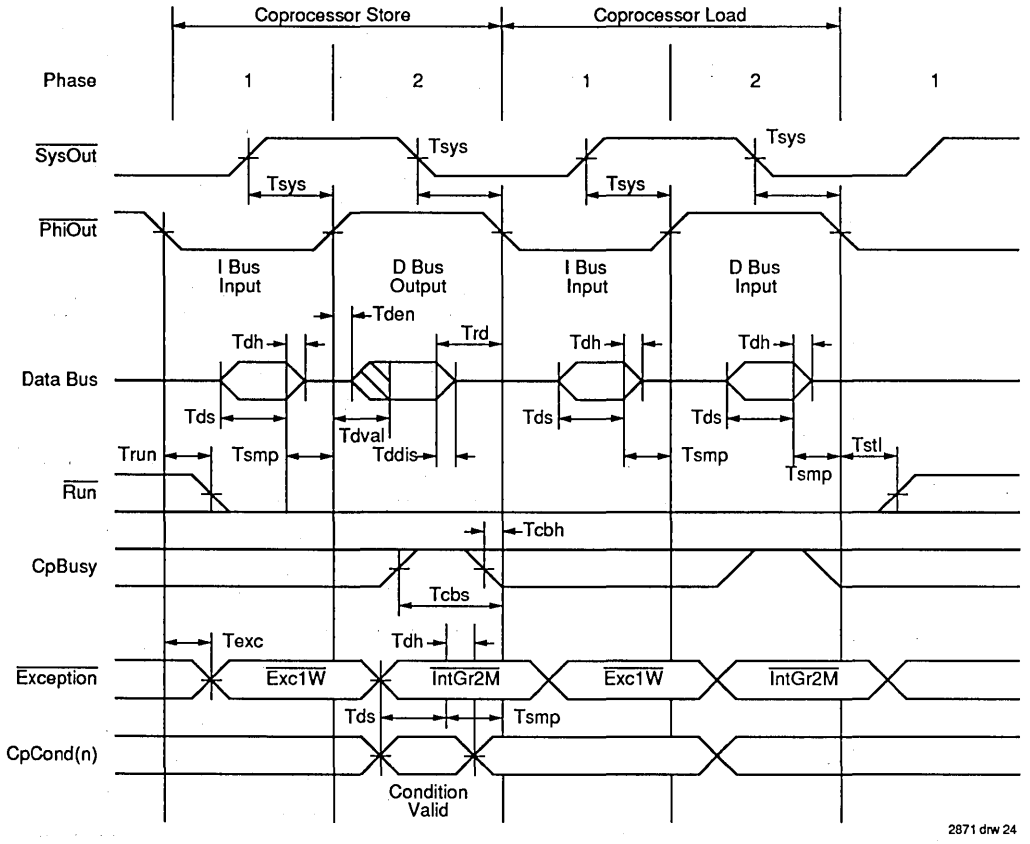


Figure 20. Coprocessor Load/Store Timing

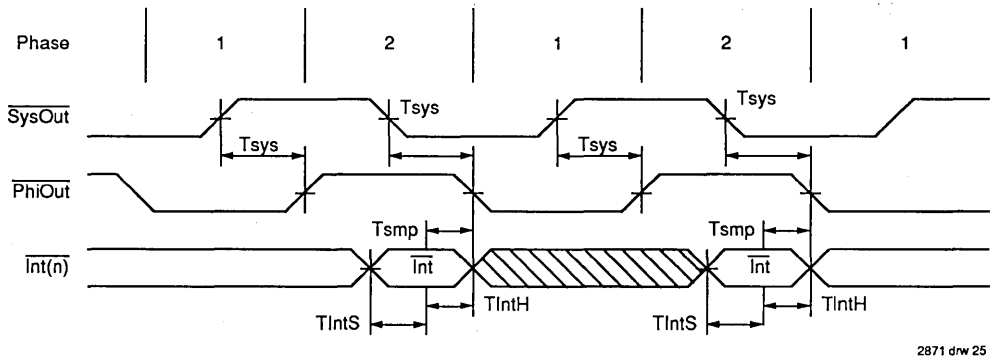


Figure 21. Interrupt Timing

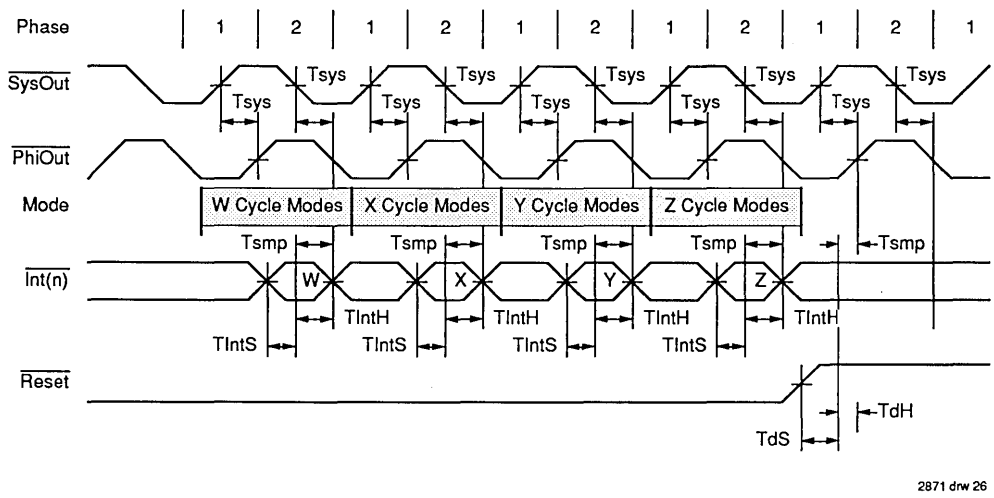
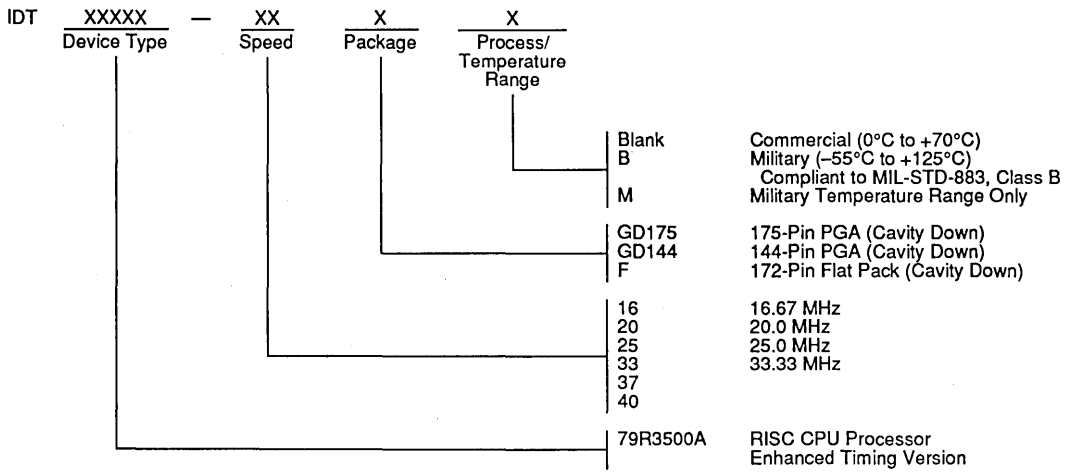


Figure 22. Mode Vector Initialization

NOTES:
1. Reset must be negated synchronously; however, it should be asserted asynchronously. Designs must not rely on the proper functioning of SysOut prior to the assertion of Reset.
2. If Phase-Lock On or R3000 Mode are asserted as mode select options, they should be asserted throughout the Reset period, to insure that the slowest coprocessor in the system has sufficient time to lock to the CPU clocks.
3. Reset is actually sampled in both Phase 1 and Phase 2. To insure proper initialization, it must be negated relative to the end of Phase 1.

ORDERING INFORMATION



2871 drw 27

AE2860-0

Integrated Device Technology, Inc.

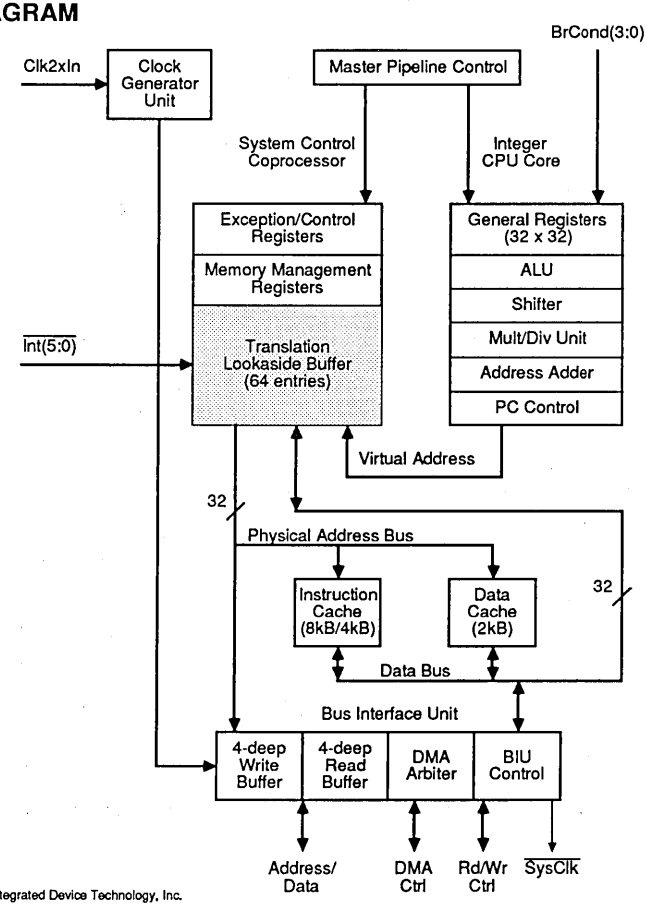
IDT79R3051 FAMILY OF INTEGRATED RISControllers™

ADVANCE INFORMATION
IDT 79R3051™, 79R3051E
IDT 79R3052™, 79R3052E

FEATURES:

- Instruction set compatible with IDT79R3000A and IDT79R3001 MIPS RISC CPUs
- High level of integration minimizes system cost, power consumption
  - — 79R3000A /79R3001 Execution Engine
  - — R3051 features 4kB of Instruction Cache
  - — R3052 features 8kB of Instruction Cache
  - — All devices feature 2kB of Data Cache
  - — "E" Versions (Extended Architecture) feature full function Memory Management Unit, including 64-entry Translation Lookaside Buffer (TLB)
  - — 4-deep write buffer eliminates memory write stalls
  - — 4-deep read buffer supports burst refill
  - — On-chip DMA arbiter
  - — Bus Interface Minimizes Processor Stalls
- Single clock input
- Direct interface to R3720/21/22 RISChipset
- 35 MIPS, over 64,000 Dhrystones at 40 MHz
- Low cost 84-pin PLCC packaging
- Flexible bus interface allows simple, low cost designs
- 20, 25, 33, and 40 MHz operation
- Complete software support
  - — Optimizing compilers
  - — Real-time operating systems
  - — Monitors/debuggers
  - — Floating Point Software
  - — Page Description Languages

R3051 FAMILY BLOCK DIAGRAM



RISController, R305x, R3051, R3052 are trademarks of Integrated Device Technology, Inc.

NOVEMBER 1990

INTRODUCTION

The IDT R3051 Family is a series of high-performance 32-bit microprocessors featuring a high level of integration, and targeted to high-performance but cost sensitive embedded processing applications. The R3051 family is designed to bring the high-performance inherent in the MIPS RISC architecture into low-cost, simplified, power sensitive applications.

Functional units were integrated onto the CPU core in order to reduce the total system cost, rather than to increase the inherent performance of the integer engine. Thus, the R3051 family is able to offer 35 MIPS of integer performance at 40 MHz without requiring external SRAM or caches.

Further, the R3051 family brings dramatic power reduction to these embedded applications, allowing the use of low-cost packaging for devices up to 25 MHz. The R3051 family allows customer applications to bring maximum performance at minimum cost.

Figure 1 shows a block level representation of the functional units within the R3051 family. The R3051 family could be viewed as the embodiment of a discrete solution built around the IDT 79R3000A or 79R3001. However, by integrating this functionality on a single chip, dramatic cost and power reductions are achieved.

Currently, there are four members of the R3051 family family. All devices are pin and software compatible: the differences lie in the amount of instruction cache, and in the memory management capabilities of the processor:

- The R3052"E" incorporates 8kB of Instruction Cache, and features a full function memory management unit (MMU) including a 64-entry fully-associative Translation Lookaside Buffer (TLB). This is the same memory management unit incorporated in the IDT 79R3000A and 79R3001.
- The R3052 also incorporates 8kB of Instruction Cache. However, the memory management unit is a much simpler subset of the capabilities of the enhanced versions of the architecture, and in fact does not use a TLB.
- The R3051"E" incorporates 4kB of Instruction Cache. Additionally, this device features the same full function MMU (including TLB file) as the R3052"E", and R3000A.
- The R3051 incorporates 4kB of Instruction Cache, and uses the simpler memory management model of the R3052.

An overview of the functional blocks incorporated in these devices follows.

CPU Core

The CPU core is a full 32-bit RISC integer execution engine, capable of sustaining close to single cycle execution rate. The CPU core contains a five stage pipeline, and 32 orthogonal 32-bit registers. The R3051 family implements the MIPS-I ISA. In fact, the execution engine of the R3051 family is the same as the execution engine of the R3000A (and R3001). Thus, the R3051 family is binary compatible with those CPU engines.

The execution engine of the R3051 family uses a five-stage pipeline to achieve close to single cycle execution. A new instruction can be started in every clock cycle; the execution engine actually processes five instructions concurrently (in various pipeline stages). Figure 2 shows the concurrency achieved by the R3051 family pipeline.

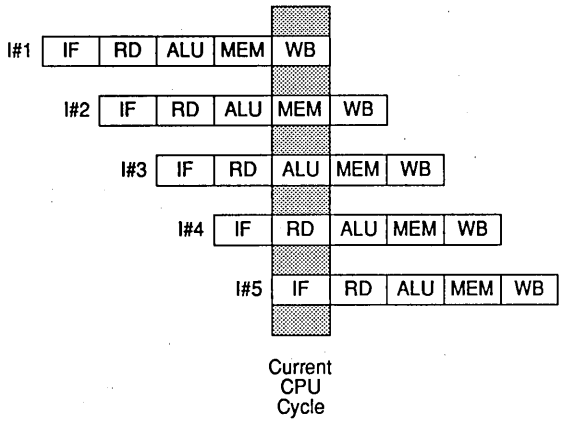


Figure 2. R3051 Family 5-Stage Pipeline

System Control Co-Processor

The R3051 family also integrates on-chip the System Control Co-processor, CP0. CP0 manages both the exception handling capability of the R3051 family, as well as the virtual to physical mapping of the R3051 family.

There are two versions of the R3051 family architecture: the Extended Architecture Versions (the R3051E and R3052E) contain a fully associative 64-entry TLB which maps 4kB virtual pages into the physical address space. The virtual to physical mapping thus includes kernel segments which are hard mapped to physical addresses, and kernel and user segments which are mapped on a page basis by the TLB into anywhere within the 4GB physical address space. In this TLB, 8 page translationss can be "locked" by the kernel to insure deterministic response in real-time applications. These versions thus use the same MMU structure as that found in the IDT 79R3000A and 79R3001. Figure 3 shows the virtual to physical address mapping found in the extended architecture versions of the processor family.

The Extended Architecture devices allow the system designer to implement kernel software to dynamically manage User task utilization of memory resources, and also allow the Kernel to effectively "protect" certain resources from user tasks. These capabilities are important in a number of embedded applications, from process control (where resource protection may be extremely important) to X-Window display systems (where virtual memory management is extremely important).

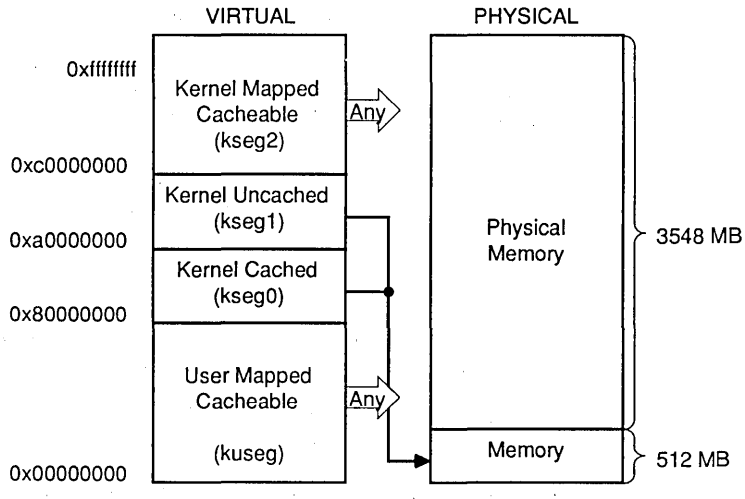


Figure 3. Virtual to Physical Mapping of Extended Architecture Versions

The base versions of the architecture (the R3051 and R3052) remove the TLB and institute a fixed address mapping for the various segments of the virtual address space. The base processors support distinct kernel and user mode operation without requiring page management software, leading to a simpler software model. The memory mapping used by these devices is illustrated in figure 4. Note that the reserved address spaces shown are for compatibility with future family members.

When using the base versions of the architecture, the system designer can implement a distinction between the user tasks and the kernel tasks, without having to execute page management software. This distinction can take the form of physical memory protection, accomplished by address decoding, or in other forms. In systems which do not wish to implement memory protection, and wish to have the kernel and user tasks operate out of a single unified memory space, upper address lines can be ignored by the address decoder, and thus all references will be seen in the lower gigabyte of the physical address space.

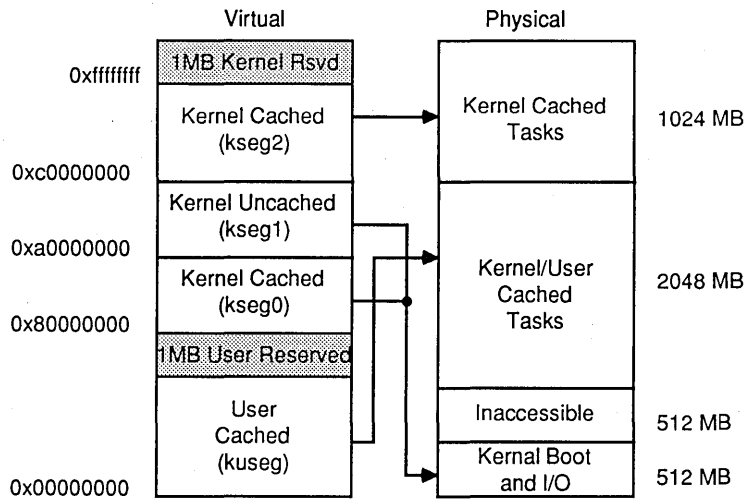


Figure 4. Virtual to Physical Mapping of Base Architecture Versions

Clock Generation Unit

The R3051 family is driven from a single input clock. On-chip, the clock generator unit is responsible for managing the interaction of the CPU core, caches, and bus interface. The clock generator unit replaces the external delay line required in R3000A and R3001 based applications.

Instruction Cache

The current family includes two different instruction cache sizes: the R3051 family (the R3051 and R3051E) feature 4kB of instruction cache, and the R3052 and R3052E each incorporate 8kB of Instruction Cache. For all four devices, the instruction cache is organized as a line size of 16 bytes (four words). This relatively large cache achieves a hit rate well in excess of 95% in most applications, and substantially contributes to the performance inherent in the R3051 family. The cache is implemented as a direct mapped cache, and is capable of caching instructions from anywhere within the 4GB physical address space. The cache is implemented using physical addresses (rather than virtual addresses), and thus does not require flushing on context switch.

Data Cache

All four devices incorporate an on-chip data cache of 2kB, organized as a line size of 4 bytes (one word). This relatively large data cache achieves hit rates well in excess of 90% in most applications, and contributes substantially to the performance inherent in the R3051 family. As with the instruction cache, the data cache is implemented as a direct mapped physical address cache. The cache is capable of mapping any word within the 4GB physical address space.

The data cache is implemented as a write through cache, to insure that main memory is always consistent with the internal cache. In order to minimize processor stalls due to data write operations, the bus interface unit incorporates a 4-deep write buffer which captures address and data at the processor execution rate, allowing it to be retired to main memory at a much slower rate without impacting system performance.

Bus Interface Unit

The R3051 family uses its large internal caches to provide the majority of the bandwidth requirements of the execution engine, and thus can utilize a simple bus interface connected to slow memory devices.

The R3051 family bus interface utilizes a 32-bit address and data bus multiplexed onto a single set of pins. The bus interface unit also provides an ALE signal to de-multiplex the A/D bus, and simple handshake signals to process processor read and write requests. In addition to the read and write interface, the R3051 family incorporates a DMA arbiter, to allow an external master to control the external bus.

The R3051 family incorporates a 4-deep write buffer to decouple the speed of the execution engine from the speed of the memory system. The write buffers capture and FIFO processor address and data information in store operations, and presents it to the bus interface as write transactions at the rate the memory system can accommodate. Figure 5 illustrates a basic write transaction for the R3051/52.

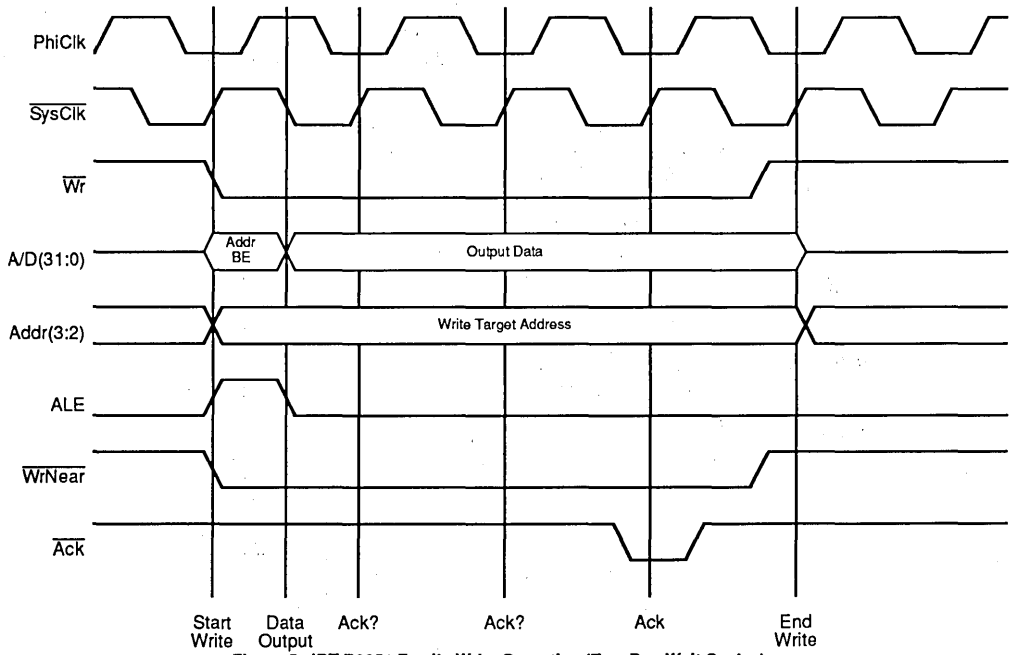


Figure 5. IDT R3051 Family Write Operation (Two Bus Wait Cycles)

The R3051/52 read interface performs both single word reads and quad word reads. Single word reads work with a simple handshake, and quad word reads can either utilize the simple handshake (in lower performance, simple systems) or utilize a tighter timing mode when the memory system can burst data at the processor clock rate. Thus, the system designer can choose to utilize page or nibble mode DRAMs (and possibly use interleaving), if desired, in high-performance systems, or use simpler techniques to reduce complexity. Figure 6 illustrates a basic single word read; figure 7 illustrates a burst block transfer. More aggressive designs could significantly reduce the number of processor stall cycles from those shown here.

In order to accommodate slower quad word reads, the R3051 family incorporates a 4-deep read buffer FIFO, so that the external interface can queue up data within the processor before releasing it to perform a burst fill of the internal caches. Figure 8 shows the action of the processor for a "throttled" quad word read. Depending on the cost vs. performance tradeoffs appropriate to a given application, the system design engineer could include true burst support from the DRAM to provide for high-performance cache miss processing, or utilize the read buffer to process quad word reads from slower memory systems.

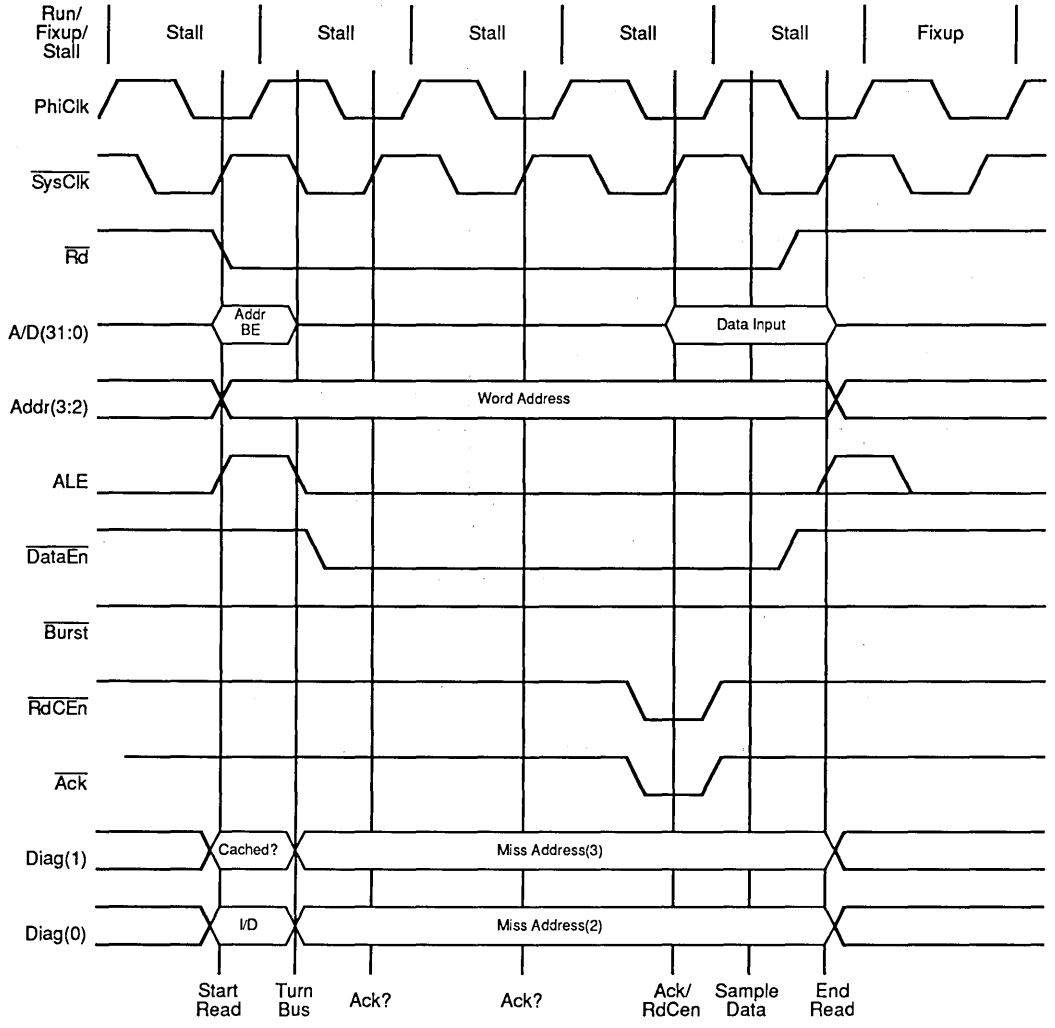


Figure 6. IDT R3051 Family Single Word Read Operation (Two Bus Wait Cycles)

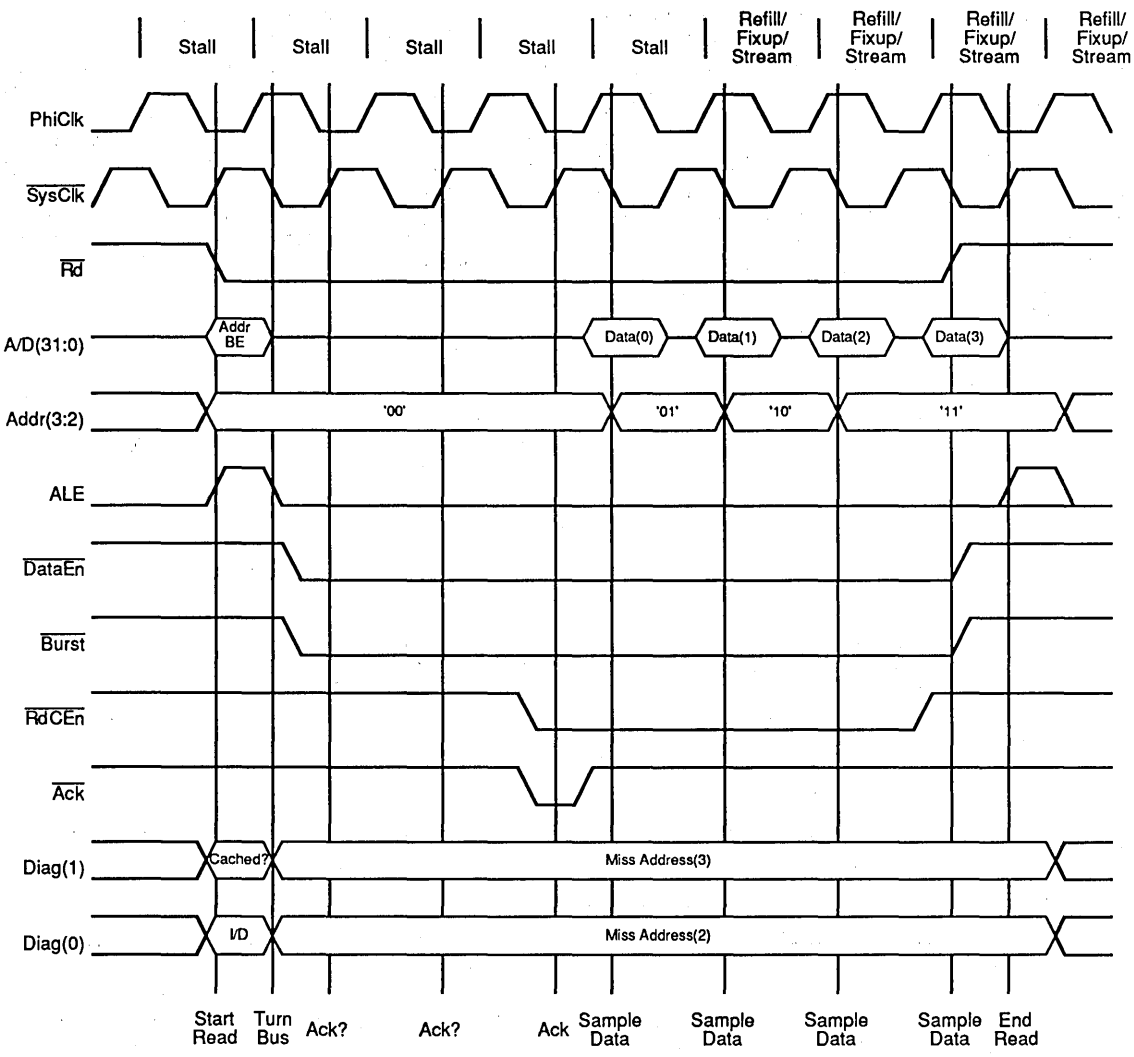


Figure 7. IDT R3051 Family Burst Read Operation (Two Bus Wait Cycles)

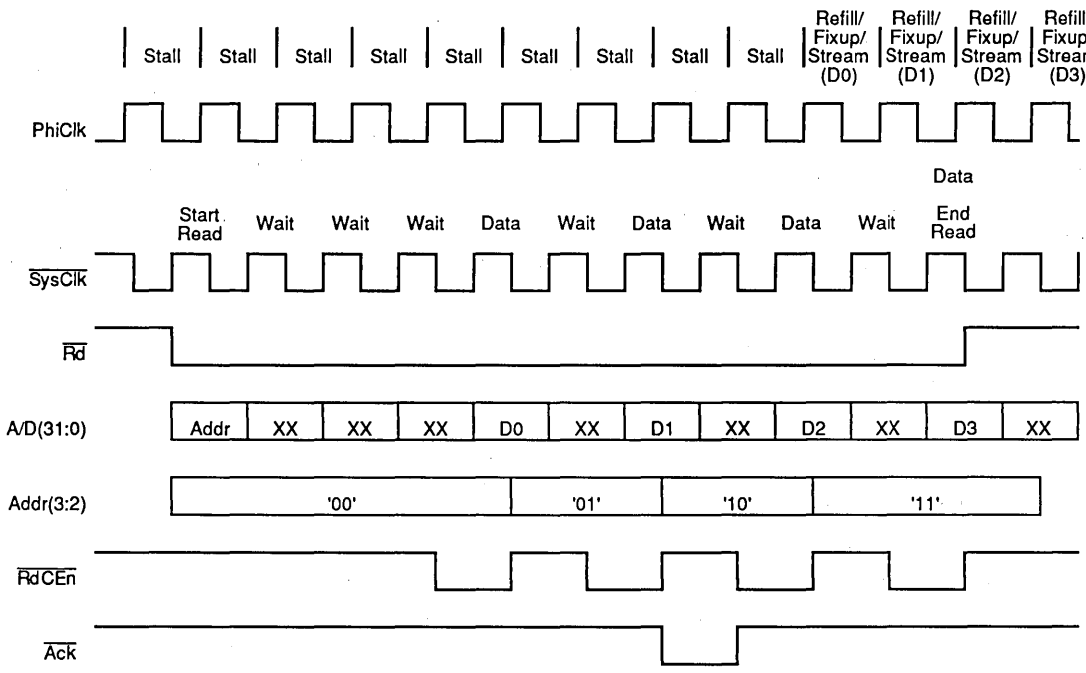


Figure 8. IDT R3051 Family Throttled Quad Read Operation (Three Bus Wait Cycles, One Bus Wait Cycle Between Words)

5

SYSTEM USAGE

The IDT R3051 family has been specifically designed to easily connect to low-cost memory systems. Typical low-cost memory systems utilize slow EPROMs, DRAMs, and application specific peripherals. These systems may also typically contain large, slow static RAMs, although the IDT R3051 family has been designed to not specifically require the use of external SRAMs.

Figure 9 shows a typical system block diagram. Transparent latches are used to de-multiplex the R3051/52 address and data busses from the A/D bus. The data paths between the memory system elements and the R3051/52 A/D bus is managed by simple octal devices. A small set of simple PALs can be used to control the various data path elements, and to control the handshake between the memory devices and the R3051/52.

Alternately, the memory interface can be constructed using the IDT R3051 family RISChipset, which includes DRAM control, data path control for interleaved memories, and other general memory and system interface control functions. These devices are described in separate data sheets. Figure 10 illustrates a simple system constructed using the R3051 family support chip set.

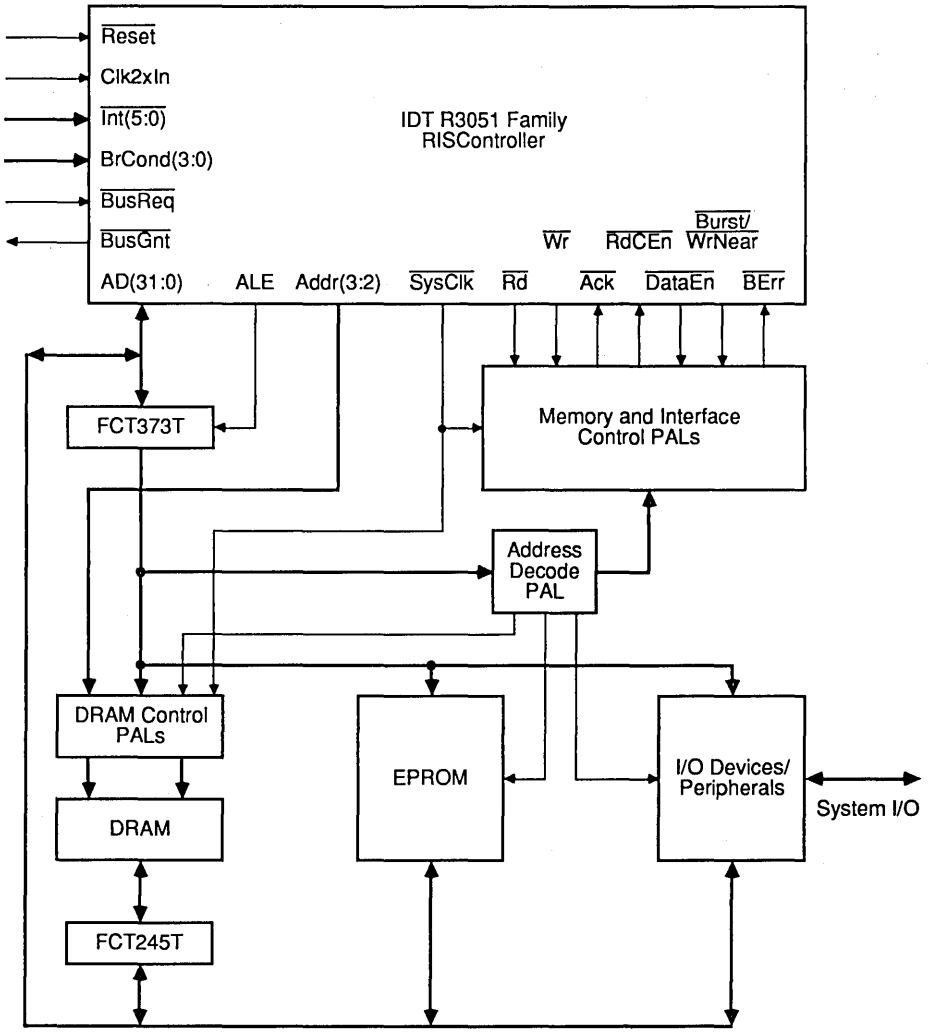


Figure 9. Typical R3051 Family Based System

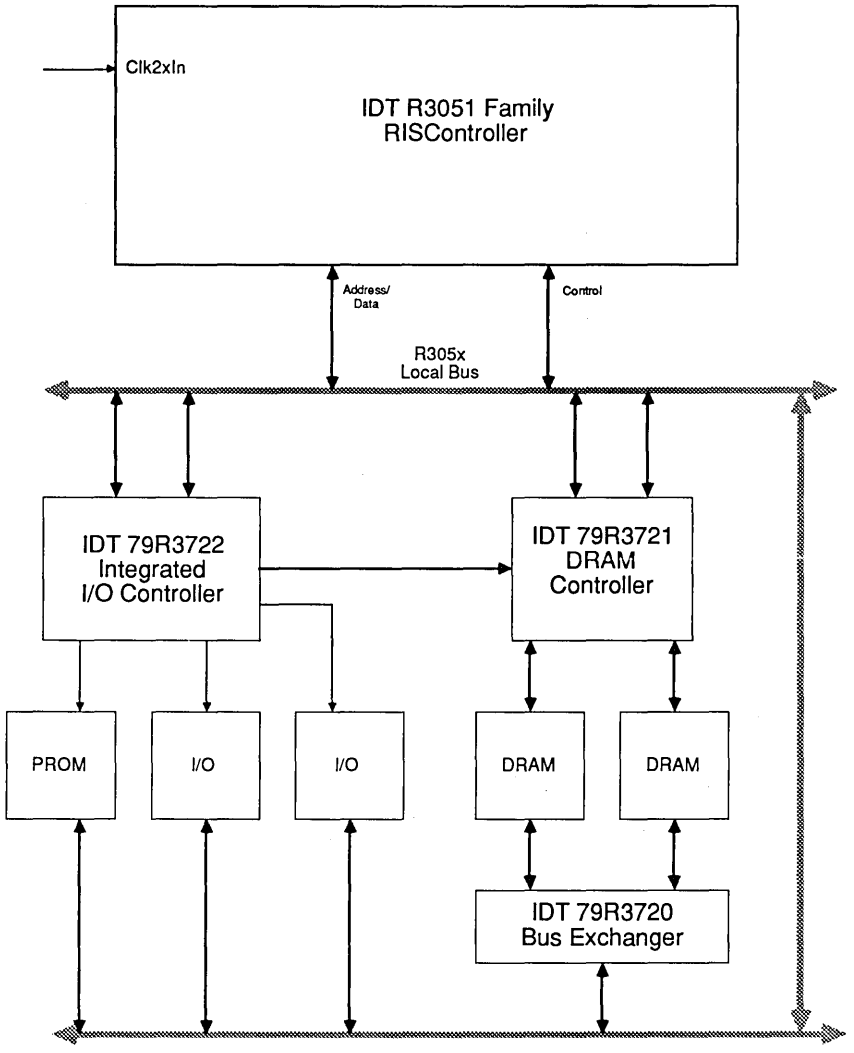


Figure 10. R3051 Family Chip Set Based System

DEVELOPMENT SUPPORT

The IDT R3051 family is supported by a rich set of development tools, ranging from system simulation tools through prom monitor support, logic analysis tools, and sub-system modules.

Figure 11 is an overview of the system development process typically used when developing R3051 family-based applications. The R3051 family is supported by powerful tools through all phases of project development. These tools allow timely, parallel development of hardware and software for R3051/52 based applications, and include tools such as:

- A program, Cache-3051, which allows the performance of an R3051 family based system to be modeled and understood without requiring actual hardware.
- Sable, an instruction set simulator.
- Optimizing compilers from MIPS, the acknowledged leader in optimizing compiler technology.
- IDT Cross development tools, available in a variety of development environments.
- The high-performance IDT floating point library software.
- The IDT Evaluation Board, which includes RAM, EPROM, I/O, and the IDT Prom Monitor.
- The IDT Laser Printer System board, which directly drives a low-cost print engine, and runs Microsoft TrueImage™ Page Description Language on top of PeerlessPage™ Advanced Printer Controller BIOS.
- Adobe PostScript™ Page Description Language, ported to the R3000 instruction set, runs on the IDT R3051 family.
- The IDT Prom Monitor, which implements a full prom monitor (diagnostics, remote debug support, peek/poke, etc.).

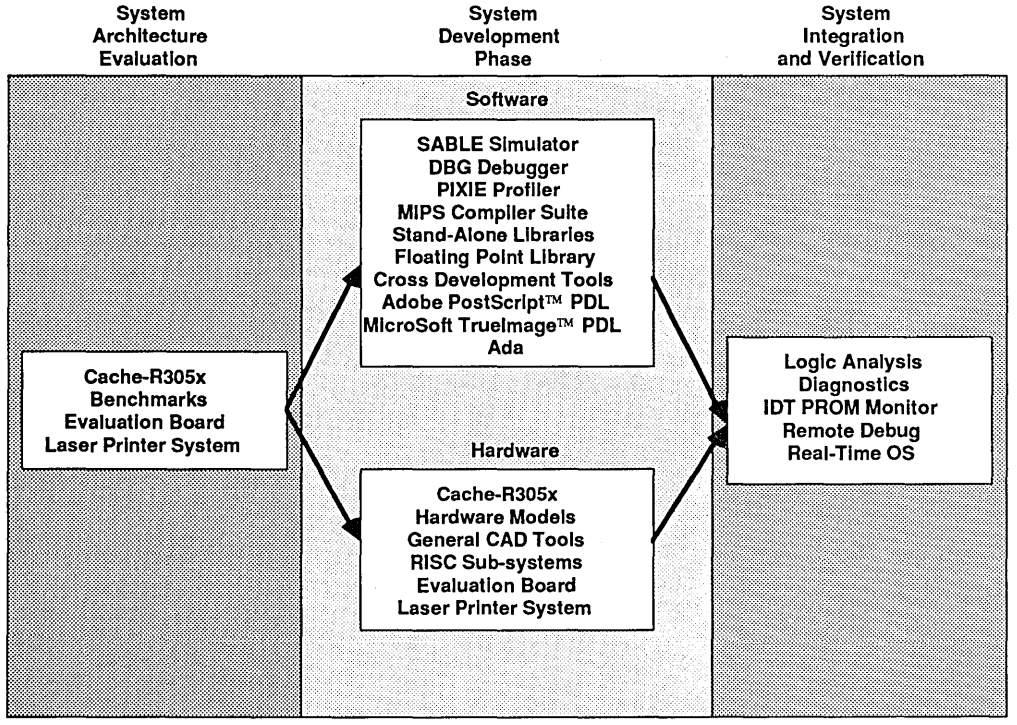


Figure 11. R3051 Family Development Toolchain

PERFORMANCE OVERVIEW

The R3051 family achieves a very high-level of performance. This performance is based on:

- **An efficient execution engine.** The CPU performs ALU operations and store operations at a single cycle rate, and has an effective load time of 1.3 cycles, and branch execution rate of 1.5 cycles (based on the ability of the compilers to avoid software interlocks). Thus, the execution engine achieves over 35 MIPS performance when operating out of cache.
- **Large on-chip caches.** The R3051 family contains caches which are substantially larger than those on the majority of today's embedded microprocessors. These large caches minimize the number of bus transactions required, and allow the R3051 family to achieve actual sustained performance very close to its peak execution rate.
- **Autonomous multiply and divide operations.** The R3051 family features an on-chip integer multiplier/divide unit which is separate from the other ALU. This allows the R3051 family to perform multiply or divide operations in parallel with other integer operations, using a single multiply or divide instruction rather than "step" operations.
- **Integrated write buffer.** The R3051 family features a four deep write buffer, which captures store target addresses and data at the processor execution rate and retires it to main memory at the slower main memory access rate. Use of on-chip write buffers eliminates the need for the processor to stall when performing store operations.
- **Burst read support.** The R3051 family enables the system designer to utilize page mode or nibble mode RAMs when performing read operations to minimize the main memory read penalty and increase the effective cache hit rates.

These techniques combine to allow the processor to achieve 35 MIPS integer performance, and over 64,000 dhrystones at 40 MHz without the use of external caches or zero wait-state memory devices.

5

PIN DESCRIPTION

Pin Name	I/O	Description
A/D(31:0)	I/O	**Address/Data:** A 32-bit time multiplexed bus which indicates the desired address for a bus transaction in one cycle, and which is used to transmit data between this device and external memory resources on other cycles. Bus transactions on this bus are logically separated into two phases: during the first phase, information about the transfer is presented to the memory system to be captured using the ALE output. This information consists of: **Address(31:4):** The high-order address for the transfer is presented. **$\overline{BE}$(3:0):** These strobes indicate which bytes of the 32-bit bus will be involved in the transfer. During write cycles, the bus contains the data to be stored and is driven from the internal write buffer. On read cycles, the bus receives the data from the external resource, in either a single word transaction or in a burst of four words, and places it into the on-chip read buffer.
Addr(3:2)	O	**Low Address (3:2)** A 2-bit bus which indicates which word is currently expected by the processor. Specifically, this two bit bus presents either the address bits for the single word to be transferred (writes or single word reads) or functions as a two bit counter starting at '00' for burst read operations.
Diag(1)	O	**Diagnostic Pin 1.** This output indicates whether the current bus read transaction is due to an on-chip cache miss, and also presents part of the miss address. The value output on this pin is time multiplexed: **Cached:** During the phase in which the A/D bus presents address information, this pin is an active high output which indicates whether the current read is a result of a cache miss.The value of this pin at this time in other than read cycles is undefined. **Miss Address (3):** During the remainder of the read operation, this output presents address bit (3) of the address the processor was attempting to reference when the cache miss occurred. Regardless of whether a cache miss is being processed, this pin reports the transfer address during this time.
Diag(0)	O	**Diagnostic Pin 0.** This output distinguishes cache misses due to instruction references from those due to data references, and presents the remaining bit of the miss address. The value output on this pin is also time multiplexed: **I/$\overline{D}$:** If the "Cached" Pin indicates a cache miss, then a high on this pin at this time indicates an instruction reference, and a low indicates a data reference. If the read is not due to a cache miss but rather an uncached reference, then this pin is undefined during this phase. **Miss Address (2):** During the remainder of the read operation, this output presents address bit (2) of the address the processor was attempting to reference when the cache miss occurred. Regardless of whether a cache miss is being processed, this pin reports the transfer address during this time.
ALE	O	**Address Latch Enable:** Used to indicate that the A/D bus contains valid address information for the bus transaction. This signal is used by external logic to capture the address for the transfer.
$\overline{DataEn}$	O	**Data Input Enable:** This signal indicates that the A/D bus is no longer being driven by the processor during read cycles, and thus the external memory system may enable the drivers of the memory system onto this bus without having a bus conflict occur. During write cycles, or when no bus transaction is occurring, this signal is negated.

PIN DESCRIPTION (Continued)

Pin Name	I/O	Description
Burst/ WrNear	O	**Burst Transfer/Write Near:** On read transactions, this signal indicates that the current bus read is requesting a block of four contiguous words from memory. This signal is asserted only in read cycles due to cache misses; it is asserted for all I-Cache miss read cycles, and for D-Cache miss read cycles if selected at device reset time. On write transactions, this output tells the external memory system that the bus interface unit is performing back-to-back write transactions to an address within the same 256 byte page as the prior write transaction. This signal is useful in memory systems which employ page mode or static column DRAMs.
Rd	O	**Read:** An output which indicates that the current bus transaction is a read.
Wr	O	**Write:** An output which indicates that the current bus transaction is a write.
Ack	I	**Acknowledge:** An input which indicates to the device that the memory system has sufficiently processed the bus transaction, and that the CPU may either advance to the next write buffer entry or process the read data.
RdCEn	I	**Read Buffer Clock Enable:** An input which indicates to the device that the memory system has placed valid data on the A/D bus, and that the processor may move the data into the on-chip Read Buffer.
SysClk	O	**System Reference Clock:** An output from the CPU which reflects the timing of the internal processor sys clock. This clock is used to control state transitions in the read buffer, write buffer, memory controller, and bus interface unit.
BusReq	I	**DMA Arbiter Bus Request:** An input to the device which requests that the CPU tri-state its bus interface signals so that they may be driven by an external master.
BusGnt	O	**DMA Arbiter Bus Grant.** An output from the CPU used to acknowledge that a BusReq has been detected, and that the bus is relinquished to the external master.
SBrCond(3:2) BrCond(1:0)	I	**Branch Condition Port:** These external signals are internally connected to the CPU signals CpCond(3:0). These signals can be used by the branch on co-processor condition instructions as input ports. There are two types of Branch Condition inputs: the SBrCond inputs have special internal logic to synchronize the inputs, and thus may be driven by asynchronous agents. The direct Branch Condition inputs must be driven synchronously.
BErr	I	Bus Error: Input to the bus interface unit to terminate a bus transaction due to an external bus error. This signal is only sampled during read and write operations. If the bus transaction is a read operation, then the CPU will take a bus error exception.
Int(5:3) SInt(2:0)	I	**Processor Interrupt:** During operation, these signals are logically the same as the Int(5:0) signals of the R3000. During processor reset, these signals perform mode initialization of the CPU, but in a different (simpler) fashion than the interrupt signals of the R3000. There are two types of interrupt inputs: the SInt inputs are internally synchronized by the processor, and may be driven by an asynchronous external agent. The other interrupt inputs are not internally synchronized. The direct interrupt inputs have one cycle lower latency than the synchronized interrupts.
Clk2xIn	I	**Master clock input:** This is a double frequency input used to control the timing of the CPU. Internally, the clock generator unit derives the four processor "2xclk" signals from this clock.
Reset	I	**Master Processor Reset:** This signal initializes the CPU. Mode selection is performed during the last cycle of reset.
Rsvd(4:0)	I/O	**Reserved:** These five signal pins are reserved for testing and for future revisions of this device. Users must not connect these pins.

5

ABSOLUTE MAXIMUM RATINGS(1, 3)

Symbol	Rating	Commercial	Military	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70 (Ambient)	–55 to +125 (Case)	°C
TBIAS	Temperature Under Bias	–55 to +125	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
VIN	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

NOTE:
1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. VIN minimum = -3.0V for pulse width less than 15ns. VIN should not exceed VCC +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

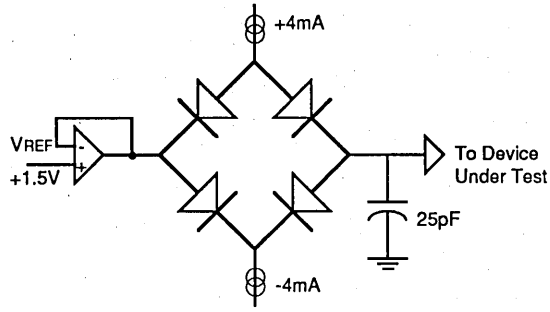
AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	Vcc
Military	-55°C to +125°C (Case)	0V	5.0 ±10%
Commercial	0°C to +70°C (Ambient)	0V	5.0 ±5%

OUTPUT LOADING FOR AC TESTING



DC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	20MHz		25MHz		33.33MHz		40MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{IH}	Input HIGH Voltage[3]		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage[1]		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage[2,3]		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage[1,2]		—	0.4	—	0.4	—	0.4	—	0.4	V
C_{IN}	Input Capacitance[4]		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance[4]		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	350	—	400	—	500	—	600	mA
I_{IH}	Input HIGH Leakage	V_{IH} = VCC	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage	V_{IL} = GND	-100	—	-100	—	-100	—	-100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	-100	100	-100	100	-100	100	-100	100	µA

NOTES:

1. V_{IL} Min. = -3.0V for pulse width less than 15ns. V_{IL} should not fall below -0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xIn and $\overline{Reset}$.
3. V_{IH} should not be held above V_{CC} + 0.5 volts.
4. Guaranteed by design.

5

AC ELECTRICAL CHARACTERISTICS (1, 2, 3)

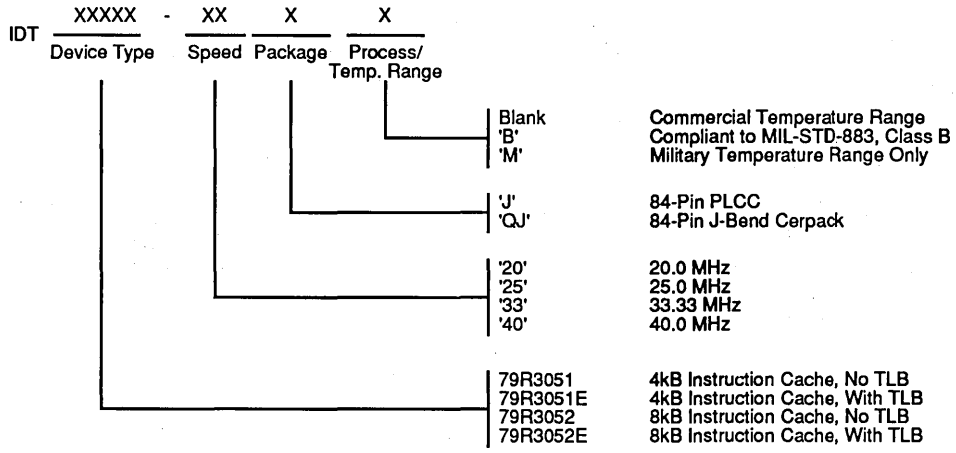
COMMERCIAL TEMPERATURE RANGE (TA = 0°C to +70°C, Vcc = +5.0V ±5%)

Symbol	Signals	Description	20MHz		25MHz		33.33MHz		40MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t1	BusReq, Ack, BusError, SInt, RdCEn, Int, BrCond, SBrCond	Set-up to SysClk rising	6	—	5	—	4	—	3	—	ns
t1a	A/D	Set-up to SysClk falling	6	—	5	—	4	—	3	—	ns
t2	BusReq, Ack, BusError, SInt, RdCEn, Int, BrCond, SBrCond	Hold from SysClk rising	4	—	4	—	3	—	3	—	ns
t2a	A/D	Hold from SysClk falling	2	—	2	—	1	—	1	—	
t3	A/D, Addr, Diag, ALE, Wr Burst/WrNear, Rd, DataEn	Tri-state from SysClk rising	—	10	—	10	—	10	—	10	ns
t4	A/D, Addr, Diag, ALE, Wr Burst/WrNear, Rd, DataEn	Driven from SysClk falling	—	10	—	10	—	10	—	10	ns
t5	BusGnt	Asserted from SysClk rising	—	8	—	7	—	6	—	5	ns
t6	BusGnt	Negated from SysClk falling	—	8	—	7	—	6	—	5	ns
t7	Wr, Rd, Burst/WrNear, A/D	Valid from SysClk rising	—	5	—	5	—	4	—	3	ns
t8	ALE	Asserted from SysClk rising	—	4	—	4	—	3	—	2	ns
t9	ALE	Negated from SysClk falling	—	4	—	4	—	3	—	2	ns
t10	A/D	Hold from ALE negated(4)	2	—	2	—	1.5	—	1.5	—	ns
t11	DataEn	Asserted from SysClk falling	—	15	—	15	—	13	—	12	ns
t12	DataEn	Asserted from A/D tri-state(4)	0	—	0	—	0	—	0	—	ns
t14	A/D	Driven from SysClk rising(4)	0	—	0	—	0	—	0	—	ns
t15	Wr, Rd, DataEn, Burst/WrNear	Negated from SysClk falling	—	7	—	6	—	5	—	4	ns
t16	Addr(3:2)	Valid from SysClk	—	6	—	6	—	5	—	4	ns
t17	Diag	Valid from SysClk	—	7	—	7	—	6	—	5	ns
t18	A/D	Tri-state from SysClk falling	—	10	—	10	—	9	—	8	ns
t19	A/D	SysClk falling to data out	—	10	—	10	—	9	—	8	ns
t20	Clk2xIn	Pulse Width High	10	—	8	—	6.5	—	5	—	ns
t21	Clk2xIn	Pulse Width Low	10	—	8	—	6.5	—	5	—	ns
t22	Clk2xIn	Clock Period	25	—	20	—	15	—	12.5	—	ns
t23	Reset	Pulse Width from Vcc valid	200	—	200	—	200	—	200	—	µs
t24	Reset	Minimum Pulse Width	32	—	32	—	32	—	32	—	tsys
t25	Reset	Set-up to SysClk falling	6	—	5	—	4	—	3	—	ns
t26	Int	Mode set-up to Reset rising	6	—	5	—	4	—	3	—	ns
t27	Int	Mode hold from Reset rising	2	—	2	—	1	—	1	—	ns
t28	SInt, SBrCond	Set-up to SysClk falling	6	—	5	—	4	—	3	—	ns
t29	SInt, SBrCond	Hold from SysClk falling	2	—	2	—	1	—	1	—	ns
t30	Int, BrCond	Set-up to SysClk falling	6	—	5	—	4	—	3	—	ns
t31	Int, BrCond	Hold from SysClk falling	2	—	2	—	1	—	1	—	ns
tsys	SysClk	Pulse Width	2*t22	2*t22	2*t22	2*t22	2*t22	2*t22	2*t22	2*t22	
t32	SysClk	Clock High Time	t22 - 2	t22 + 2	t22 - 2	t22 + 2	t22 - 1	t22 + 1	t22 - 1	t22 + 1	ns
t33	SysClk	Clock Low Time	t22 - 2	t22 + 2	t22 - 2	t22 + 2	t22 - 1	t22 + 1	t22 - 1	t22 + 1	ns
tderate	All outputs	Timing deration for loading over 25pf(4, 5)	—	0.5	—	0.5	—	0.5	—	0.5	ns/ 25pF

NOTES:

1. All timings referenced to 1.5 Volts.
2. All outputs tested with 25 pF loading.
3. The AC values listed here reference timing diagrams contained in the R3051 Family Hardware User's Manual.
4. Guaranteed by design.
5. This parameter is used to derate the AC timings according to the loading of the system. This parameter provides a deration for loads over the specified test condition; that is, the deration factor is applied for each 25 pF over the specified test load condition.

ORDERING INFORMATION



5

Integrated Device Technology, Inc.

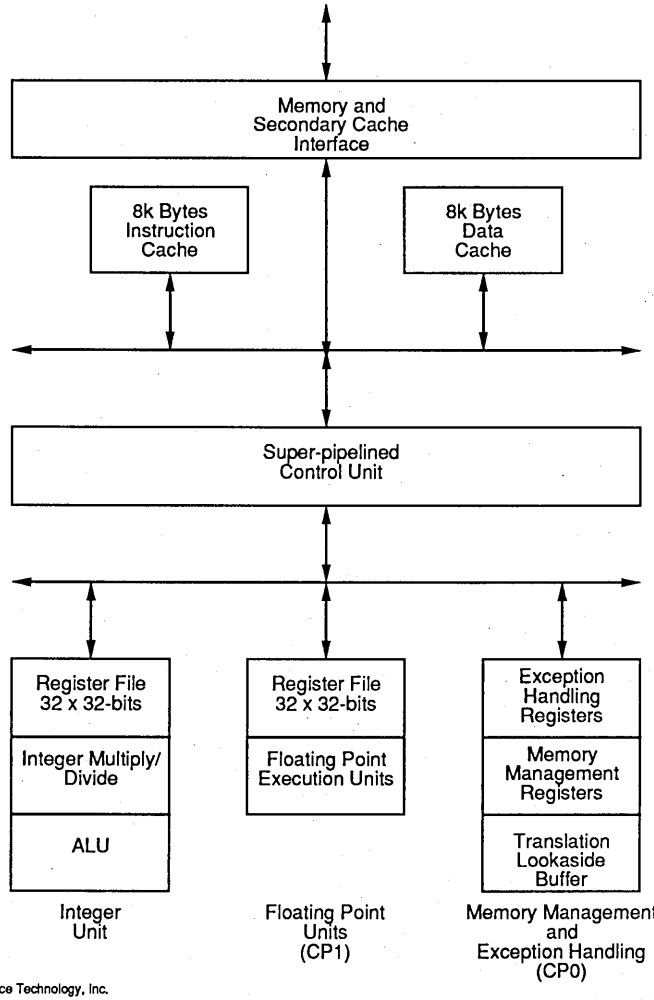
THIRD GENERATION MIPS RISC PROCESSOR

ADVANCE INFORMATION
IDT79R4000

FEATURES:

- High-Performance, Highly Integrated CPU
- Fully Binary Compatible with R2000, R3000 CPUs
- Capable of over 50 VAX MIPS sustained system performance
- High-level of performance
  - — Utilizes super-pipelining to exploit 2-level instruction level parallelism with no issue restrictions
  - — Balanced integer and floating point performance
  - — 64-bit floating point extensions
  - — Multi-processing support
- High-Level of Integration
  - — RISC Integer Unit
  - — IEEE Compatible Floating Point Units
  - — Memory Management Unit
  - — 8kB Instruction Cache
  - — 8kB Data Cache
  - — Direct control of optional secondary cache
  - — Extensive multi-processing support

BLOCK DIAGRAM



CEMOS is a trademark of Integrated Device Technology, Inc.

NOVEMBER 1990

DESCRIPTION:

The R4000 is the third generation of MIPS RISC technology, continuing MIPS track record as the performance leader and establishing a new performance standard for the 1990's.

The R4000 maintains full binary compatibility with applications executing on the R2000 and R3000 MIPS RISC CPUs (also available from IDT) and IDT's RISController™ family, while achieving substantially higher performance. The key to this performance standard is both the architecture/implementation of the processor, and the level of integration achieved in a single chip. The R4000 contains both a high-performance execution core (integer and floating point) as well as sufficient memory bandwidth (large on-chip caches) to keep the execution engine running. The on-chip resources are complemented by a direct interface to an optional secondary cache, and also by multi-processing support, allowing the system designer to further increase memory bandwidth to the processor, and to operate numerous processors together to increase overall computational power. This balanced architectural approach allows R4000 based systems to achieve a wide range of price performance goals.

Keys to Performance

All microprocessors are governed by the same basic performance equation: the time required to perform a given task is the product of the number of instructions required to execute the task with the <average> time required to complete an instruction. MIPS optimizing compilers, and the MIPS RISC architecture, serve to minimize the first term in the product. The R4000 maintains the same focus on compiler technology as an extension of the CPU architecture as did the earlier generations of MIPS processors.

The remaining term in the performance equation is the average amount of time required to execute instructions. The R4000 is designed to exploit 2-level instruction parallelism, thus being able to retire 2 instructions per clock cycle (sustained). Further, the architecture, and the level of integration, allow substantially faster clock rates to be used. The combination of fast clock rates, and multiple instructions per clock cycle, minimize the average time per instruction.

Exploiting Instruction Level Parallelism

There are a number of techniques available to achieve multiple instructions per clock cycle. The two discussed most frequently are super-scalar, and super-pipelined architectures. These machines attempt to initiate multiple instructions per clock cycle, as long as there are no data dependencies between the instructions. Thus, it is up to the CPU (rather than the compilers or programmer) to detect and exploit this "instruction level parallelism" to increase performance.

Superscalar machines attempt to run multiple instructions in distinct pipelines. In order to accomplish this, execution resources must be replicated in each pipeline. Further, significant logic must exist between the pipelines to insure that data dependencies amongst multiple instructions are resolved properly, and to insure that exceptions are detected and handled precisely. Figure 2 illustrates a theoretical superscalar machine of degree 2.

Superpipelined machines attempt to initiate multiple instructions per clock cycle, sequentially. In order to do this, execution units in the basic machine pipeline which have long latencies (require a long time to complete their operation) must be pipelined, so that multiple instructions can be executing simultaneously (although sequentially) in those units. In order to achieve high performance, the speed of the individual pipestages is higher than in equivalent superscalar implementation. However, each pipestage is significantly less complex than the super-scalar equivalent, allowing these higher speeds to be achieved. Figure 3 shows the equivalent super-pipelined machine of degree 2 (not intended to represent R4000 pipeline).

Studies have shown that architecturally, superscalar and superpipelined are duals of each other: that is, each approach is roughly equivalent in its ability to exploit instruction level parallelism. Differences in performance will then be related to the actual implementation of those techniques in a given processor, constrained by the current semiconductor technology. Compromises and trade-offs which may reduce effectiveness from the theoretical machine include:

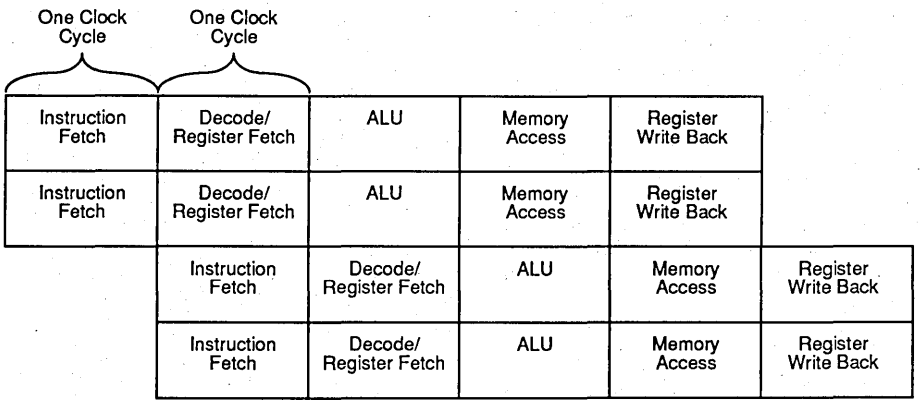


Figure 2. Superscalar RISC Pipeline

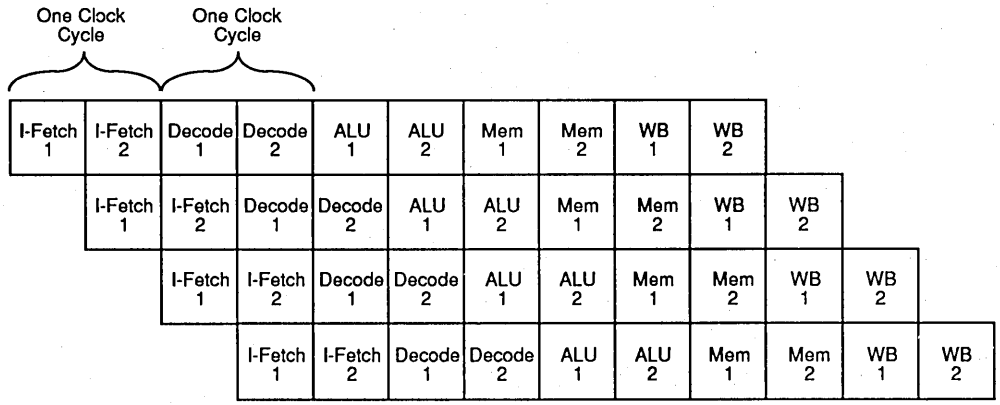


Figure 3. Superpipelined RISC Pipeline

- **Issue Restrictions:** Due to the complexity and implementation cost of replicating execution units, and checking for data dependencies between parallel pipelines, many super-scalar machines institute restrictions in the types of operations that may be initiated in parallel. This degrades the chip from the theoretical performance of a true super-scalar machine.
- **Clock Frequency:** the complexity of an implementation affects the clock frequencies achievable. A machine designed for instruction level parallelism may become so complex that the clock frequency is adversely affected. This can be a greater factor in multi-chip implementations, where significant speed is lost in bringing signals from one packaged part to another across a PC board.
- **Memory Bandwidth:** a high-performance microprocessor needs substantial memory bandwidth to achieve its performance potential. If the implementation is too complex, there may not be enough room for adequate caches to keep the execution engine fed. A good implementation is able to integrate sufficient cache memory to allow the execution engine to frequently operate at its peak performance rating.
- **Compiler technology:** the compiler technology must be capable of generating efficient code for the execution engine. In the case of a machine with issue restrictions, complex peephole optimizations may be required to maximize parallel operation. These optimizations are generally outside the realm of other traditional optimizations required for more general (less restrictive) machines.
- **Level of parallelism exploited.** A tradeoff can be made between machines with significant issue restrictions but high peak parallelism and machines with few or no issue restrictions but less peak parallelism. Numerous studies have shown that a machine capable of exploiting two level instruction parallelism, with no issue restrictions, will outperform a machine capable of exploiting 4-level parallelism but which implements significant issue restrictions. Further, the amount of parallelism exploited by the machine may cause tradeoffs in other areas, such as amount of cache or clock frequency.

Based on these constraints, the architects of the R4000 have implemented a super-pipelined execution engine to exploit 2-level instruction parallelism, with no issue restrictions, and with substantial primary instruction and data caches. The R4000 is thus capable of the raw execution speed required to achieve high-performance, and supplies sufficient bandwidth from its primary caches to minimize main memory cycles. Finally, the R4000 is able to benefit from the strength of the MIPS optimizing compiler technology, without requiring an arbitrarily complex peephole scheduler.

Level of Integration:

The R4000 brings all of the execution resources necessary for a high-performance computing system into a single chip. These resources include:

- **A High Performance Execution Engine.** The R4000 utilizes a super-pipelined execution engine, while maintaining full binary compatibility with the R2000/R3000.
- **Full featured MMU.** The R4000 integrates memory management and exception handling facilities on-chip as the system control co-processor (CP0), thus not requiring an external MMU device.
- **High Performance Floating Point Accelerator.** The R4000 integrates single and double precision floating point on-chip, as co-processor 1.
- **Large primary caches:** the R4000 integrates large (8k bytes each) instruction and data caches on-chip. These large caches allow the execution resources to operate at peak rates through substantial amounts of the application, resulting in high actual system performance, not just peak native MIPS.
- **Direct support for optional secondary cache.** The R4000 incorporates the ability to implement an external secondary cache, to further increase processor bandwidth. This is especially important in multi-processing systems.
- **Multi-processing support.** The R4000 provides the support necessary to implement high-performance, multi-processing systems.

APPLICATIONS

The R4000 extends the performance range served by the MIPS architecture into higher levels of performance. The R4000 provides a high-performance migration path to those applications currently served by devices such as the R3000, R3001, and R3051.

The MIPS RISC architecture has found widespread acceptance in a number of applications. These include:

- **High-performance multi-processing systems.** Further computational throughput can be achieved by implementing multiple R4000 in a single system, as illustrated in figure 4. MIPS RISC is already well represented in multi-processing applications, including systems from Silicon Graphics, Stardent Computer, and Digital Equipment Corporation. The R4000 allows these systems to implement even higher performance in each CPU, increasing overall system capability.

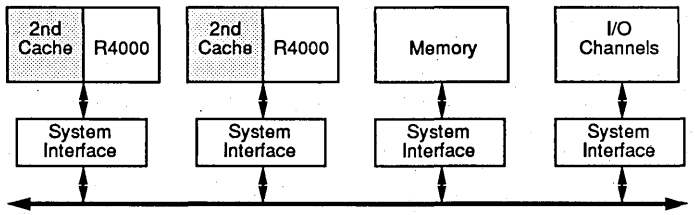


Figure 4. R4000-Based Multi-Processing System

- **Real-time systems.** The Joint Integrated Avionics Working Group (JIAWG) committee has selected the MIPS RISC architecture as a standard for military avionics. The R4000 allows these real-time applications to benefit from high integration and CPU performance.
- **Embedded computing systems.** The MIPS RISC architecture has won designs in a number of embedded systems applications, including laser printers, graphics systems, and data communications. A typical high-performance embedded system built around the R4000 is illustrated in figure 5.

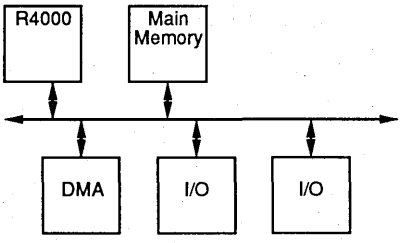


Figure 5. R4000-Based Embedded System

5

- **Desktop workstations.** MIPS RISC is a leading architecture in UNIX™ based workstations, from vendors such as MIPS, Digital Equipment Corporation, and Silicon Graphics. The R4000 extends the performance range achievable in a desktop environment, while minimizing chip count (and thus real estate, cost, and power consumption) as illustrated in figure 6.

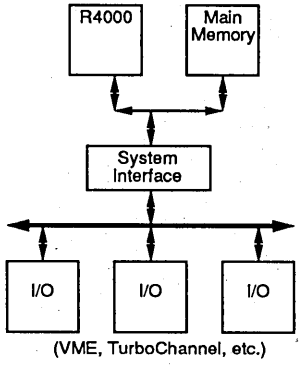


Figure 6. R4000-Based Desktop Workstation

- **Deskside server systems.** The R4000 is also capable of supporting high performance server systems, such as systems built by MIPS and Digital Equipment (around the R3000) today. Implementing a secondary cache, and a larger I/O and main memory system, extends the basic UNIX system to a high-performance server system, as shown in figure 7.

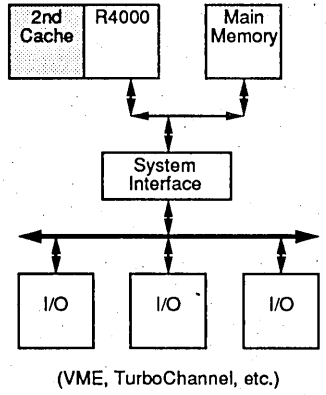


Figure 7. R4000-Based Deskside Server System

ADDITIONAL INFORMATION

Additional information on the R4000 is available from IDT. Please contact your local sales representative for additional information on this product.

RISC SUPPORT COMPONENTS

A RISC microprocessor is an important, but not self-sufficient, element of a high-performance general or embedded computing system. Equally important is the memory system (both cache and main memory) and the I/O interface to the execution core.

To simplify the task of building these high-performance subsystems, IDT produces a wide variety of support chips and building block devices. These chips range from general purpose devices such as fast static RAM and high-performance logic (used with many processor families) , to specialized devices used in only certain types of applications (such as the IDT LaserFIFO, used in laser printer systems) and devices designed to work with only a specific processor family.

Generic building block devices include SRAMs, with densities from 16KB to 1MB and access times as low as 7ns, as well as high-speed logic devices such as the FCT-T family.

Devices specifically developed for RISC systems include the RISChipset™ — 3720 Bus Exchanger, 3721 DRAM Controller and the 3722 I/O Controller. These components facilitate design of systems based upon the R3051/52 controller family. The DRAM and I/O controllers have direct bus interface to the 3051/52.

The R3020 Write Buffer enhances the performance of R3000 systems by allowing the processor to perform write operations at full clock speeds instead of resorting to time-consuming CPU stall cycles. The memory can then retire the data at a slower rate. The R32xx family of read/write buffers includes the memory read capability, enabling the use of slower main memory without impacting system performance.

By providing these system solutions as building blocks, IDT allows its customers the maximum flexibility in achieving their price performance goals while minimizing time-to-market, real estate and complexity of the end system.

This section of the data book contains some selected devices which have either been specifically designed for particular RISC processors or found to be exceptionally useful in these high-performance systems.

TABLE OF CONTENTS

Integrated Device Technology, Inc.

BUS EXCHANGER FOR R3051 FAMILY

ADVANCE INFORMATION
IDT79R3720

FEATURES:

- Direct Interface to R3051 Family RISChipSet™
  - — R3051™ Family of Integrated RISController™ CPUs
  - — R3721 DRAM Controller
  - — R3722 I/O Interface Controller
- Interfaces a single CPU bus to interleaved or banked memory systems
- Data path for read and write operations
- Low noise outputs
- Supports R3051 family systems from 20 to 33MHz
- Simplifies data path design in high-performance memory systems
- 3-Bus Architecture
  - — One CPU Bus
  - — Two (interleaved or banked) memory busses
  - — Each bus independently latched to support asynchronous operation
- 68-Pin PLCC Package
- High-performance CEMOS™ technology

DESCRIPTION:

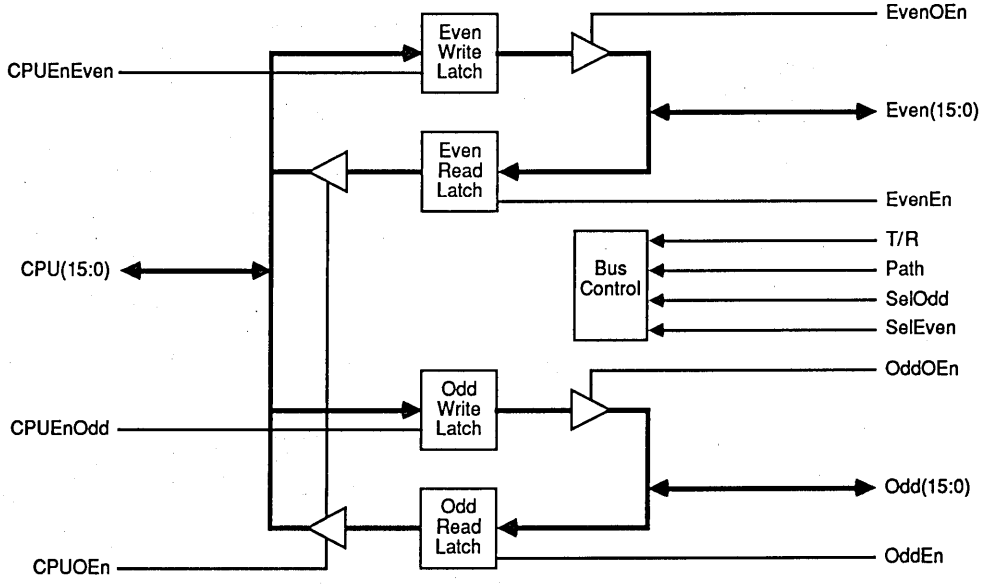
The R3720 Bus Exchanger is designed to provide data path support in an R3051 family system utilizing interleaved or banked memory techniques. The Bus Exchanger is responsible for interfacing between the CPU A/D bus (CPU address/data bus) and multiple memory data busses.

Thus, the Bus Exchanger uses a three bus architecture, with control signals suitable for simple transfer between the CPU bus and either memory bus. The Bus Exchanger features independent read and write latches for each memory bus, thus supporting a variety of memory strategies.

The bus exchanger can be used as a simple transceiver, passing data between the single CPU bus and the pair of memory busses. Alternately, data from any of the three ports can be latched by the Bus Exchanger, to free the sending port (memory, during reads, and the CPU, during writes) while the receiving port processes the transfer at its own rate.

The difference in operation is accomplished through the use of a simple set of control signals. These signals include independent latch enables for all three ports, signals to indicate the direction of transfer (read or write) and which memory port is involved, and other signals which can be used to force the ports to operate as either a transceiver or a true latch.

BLOCK DIAGRAM



6

CEMOS, RISController, R305x, R3051, R3052 are trademarks of Integrated Device Technology, Inc.

NOVEMBER 1990

USE AS PART OF THE R3051 FAMILY CHIPSET

When used with an R3051 family CPU, a pair of bus exchangers are typically used as illustrated in figure 2.

The bus exchanger is typically used as an integrated transceiver in R3051 family applications; that is, the latches are held "open" through the transfer. In such an application, the single bus exchanger replaces 8 basic transceivers plus logic to coordinate data flow between the paths.

The system memory model determines the mapping between processor addresses and memory ports. In a system with two non-interleaved banks of memory, a high-order address bit from the processor determines which memory port is being accessed. In an interleaved memory system, Addr(2) is used to alternate between two banks of memory. Both these cases are handled by the R3721 DRAM controller for the R3051 family.

The R3721 DRAM controller for the R3051 family uses the bus exchanger as a simple set of transceivers. The R3721 directly controls the inputs of the bus exchanger, during both reads and writes.

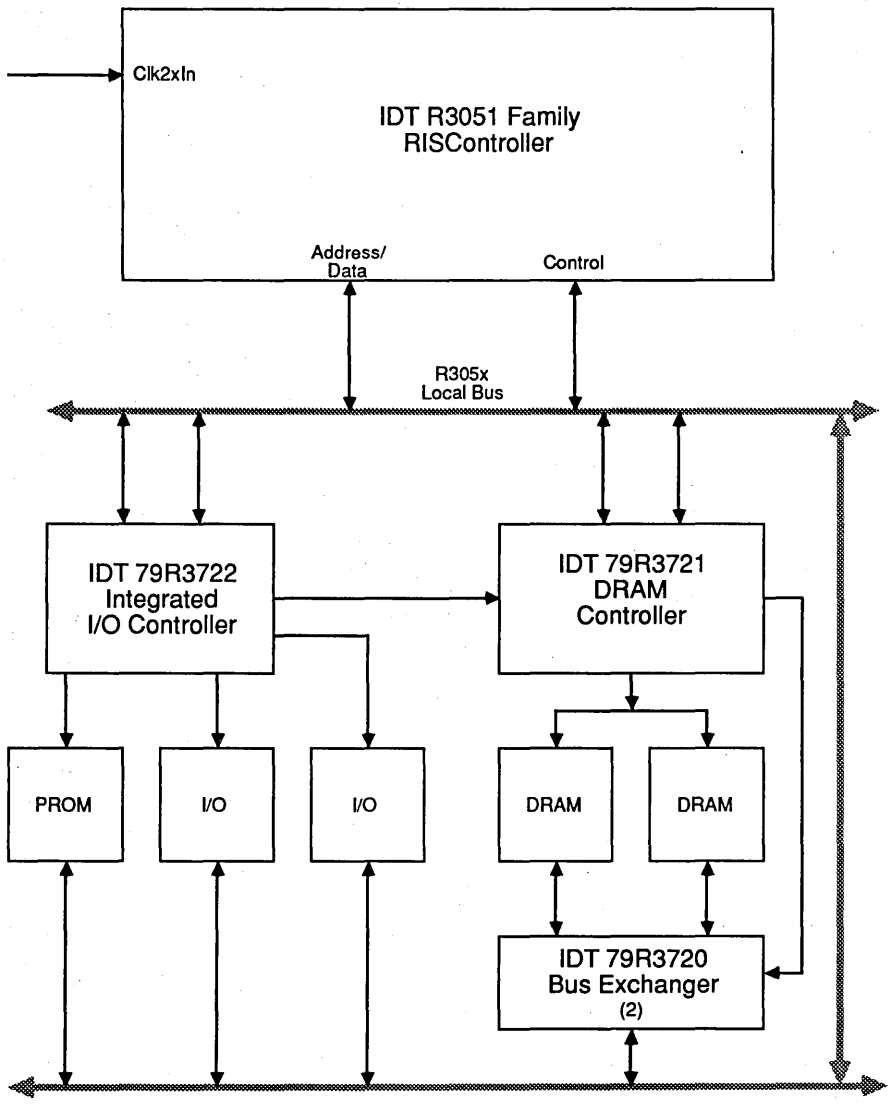


Figure 2. Bus Exchanger Used in R3051 Family System

Integrated Device Technology, Inc.

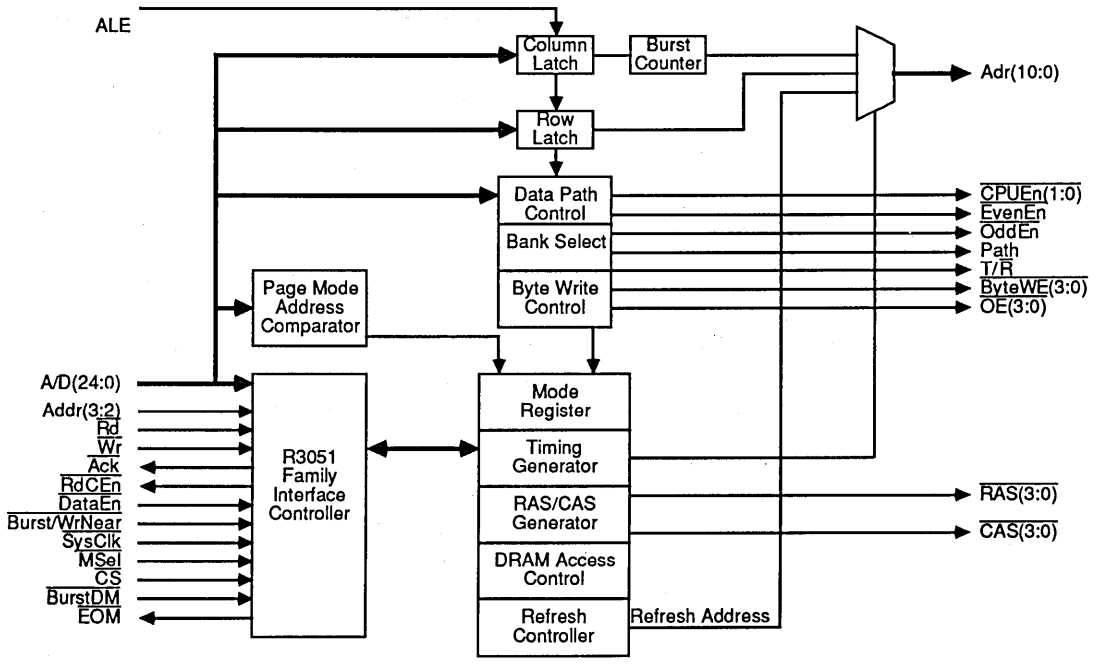
DRAM CONTROLLER FOR R3051 FAMILY

ADVANCE INFORMATION
IDT79R3721

FEATURES:

- Direct Interface to R3051™ Family RISController™ Chip Set
  - — R3051 Family Integrated RISController CPU
  - — R3720 Bus Exchanger
  - — R3722 I/O Interface Controller
- Directly drives DRAM address and control signals
  - — Directly drive 36 DRAM devices or multiple SIMM modules
  - — High output capability requires no external drivers for DRAM address or control signals
  - — Replicated control signals allows control of multiple memory banks
  - — Directly drives Bus Exchanger or 74FCT245 data path buffers
- Low noise outputs with built-in series resistance for direct drive of DRAMS (large capacitive load)
- 20, 25, and 33 MHz Operation
- Simplifies system design by eliminating glue logic/PALs
- High performance from low cost DRAMs
  - — Programmable timing model for 60 - 120 ns DRAMs
  - — Supports Page Mode read and writes using on-chip Page Detection
  - — Supports R3051 Family cache burst refill
  - — DMA interface for burst DMA read or write accesses
- Supports Multiple Common DRAM Configurations
  - — 1MB to 16 MB
  - — 256kx4 through 4Mx1 DRAMs
  - — 1, 2, or 4 Banks of DRAMs
  - — Non-interleaved or Dual Interleaving Configurations
  - — Page Mode, or Static Column Mode Accesses
  - — CAS-before-RAS or RAS-only refresh timing control on-chip
- Cascadeable to allow multiple DRAM controllers per system
- Low noise TTL level outputs for CPU interface
- High-performance CEMOS™ technology

BLOCK DIAGRAM



CEMOS, RISController, R305x, R3051, R3052 are trademarks of Integrated Device Technology, Inc.

NOVEMBER 1990

DESCRIPTION:

The R3721 DRAM controller provides a direct interface between the R3051 family CPU and DRAM-based main memory. The R3721 requires no PALs, buffers, or drivers to interface to the CPU or to the DRAM devices. It is capable of being used either with the R3722 I/O Controller or in applications which perform I/O functions using discrete or ASIC-based approaches. The R3721 DRAM controller thus dramatically simplifies the problem of interfacing DRAMs to the R3051 family.

Operation Overview

The R3721 accepts a memory decode input from an external address decoder (for example, the R3722) and performs the full data transfer requested. Using an external input to initiate transfers allows multiple R3721's to be cascaded in large memory systems.

The R3721 allows a number of different memory configurations to be used, and a number of different speeds of DRAM devices to be used, at any given CPU speed. The various options are selected by performing a write to the internal mode register of the R3721. Options in the memory configuration include: use of 256kx4 through 4Mx1 DRAM organizations; single, banked, or interleaved memory configurations; DRAM speeds of 60 - 120ns; and other options relating to DRAM refresh, use of static column mode, etc. Design decisions on which model best serves the application will be based upon the cost, performance, and main memory size requirements of the system.

The R3721 performs reads and writes, as requested by the R3051 (or other bus master, in the case of systems which include DMA). The R3721 is capable of performing the single or quad word reads requested by the processor, as well as the buffered write (either near or not near in the DRAM page). Burst reads can be processed as either true burst, or throttled reads, depending on the memory configuration and speed. Refer to the R3051 Family data sheet or Hardware User's Guide for more information on the types of transfers performed by an R3051-family CPU.

An operation is initiated when the R3721 is selected externally for a transfer. The R3721 then monitors the input control signals of the R3051 interface bus, and performs the indicated transfer. The R3721 performs the entire handshake required by the R3051 family, including the use of $\overline{\text{RdCEn}}$ and $\overline{\text{Ack}}$ to control the transfer of data between the CPU and the DRAM. In addition to performing the control interface between the R3051 bus interface and the DRAM (including addressing), the R3721 also manages the data path between the DRAMs and the R3051 family CPU, by directly controlling either a 74FCT245-type transceiver device (for single or simple banked designs) or an R3720 Bus Exchanger (for interleaved or banked memory systems).

APPLICATIONS

The R3721 is designed to interface directly to an R3051 family CPU. Additionally, the R3721 works seamlessly with the R3720 Bus Exchanger (in interleaved memory systems) and the R3722 Integrated I/O controller. Figure 2 illustrates the construction of an R3051 family systems using this chip set.

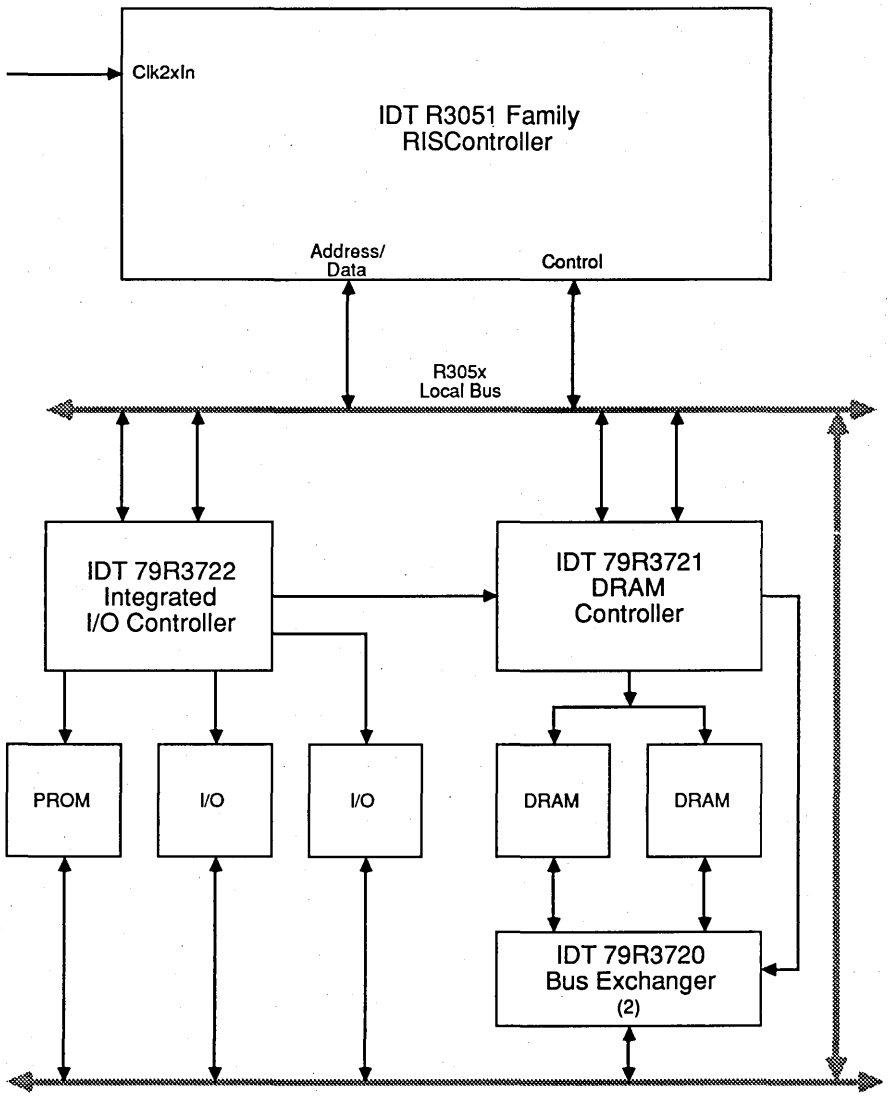


Figure 2. Use of R372x RISChipset in R3051 Family System

Integrated Device Technology, Inc.

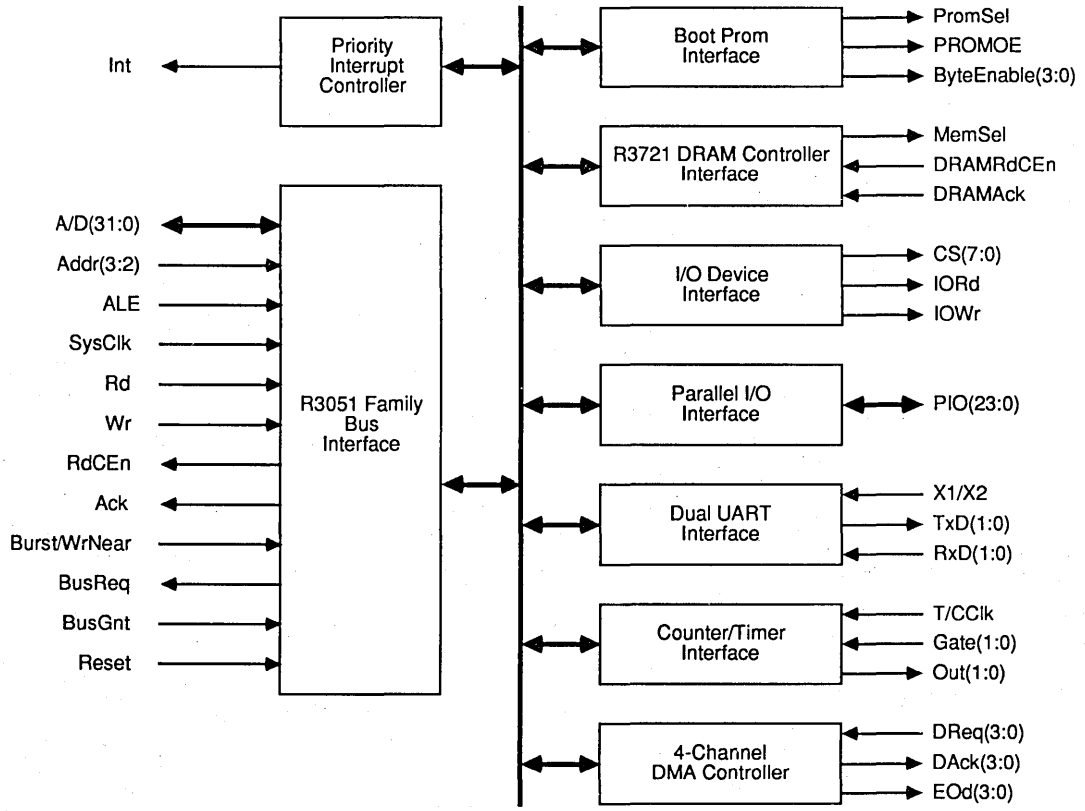
I/O INTERFACE CONTROLLER FOR R3051 FAMILY

ADVANCE INFORMATION
IDT79R3722

FEATURES:

- Direct Interface to R305x RISController™ RISChipSet™
  - — R3051 Family of Integrated RISController CPUs
  - — R3721 DRAM Controller
  - — R3720 Bus Exchanger
- Supports R3051 family systems from 20 to 40MHz
- Priority Interrupt Controller for On-Chip Peripherals
  - — Nine Internal Interrupt Requests
  - — Interrupt Pending Register
  - — Each interrupt independently maskable
- Three timer/counters with programmable counter modes
- Programmable parallel I/O port
- Boot interface supports byte-wide or 32-bit boot prom
- Dual UART
  - — Full Duplex support
  - — Programmable data formats
  - — Programmable baud rates to 19.2K baud
- High-performance DMA channels
- Interfaces between R305x and external I/O devices
  - — Address space decoding into 8 I/O segments
  - — Programmable wait state generation per I/O segment
- 132-Pin PQFP Package

BLOCK DIAGRAM



DESCRIPTION:

The R3722 is designed to link an R3051 family CPU to the various memory and I/O resources of the target application. In addition to performing the control functions required in the system, the R3722 also incorporates many of the simple peripheral functions found in typical embedded systems: parallel I/O, serial I/O, and counter/timers. Finally, the R3722 incorporates DMA channels, designed to move data between the various resources under the joint control of the R3051 and the R3722.

The R3722 is really designed to eliminate the PAL and glue logic found in many embedded systems. The R3722 provides the functions typically performed by PALs, such as: address decoding; wait state generation; synchronization; and peripheral control.

Note that the R3722 does not directly perform DRAM conttrol; rather, it decodes references to DRAM, and provides an output signal to an external DRAM controller (such as the R3721). This is an appropriate trade-off, allowing the embedded sytem designer maximal flexibility in interfacing to the memory type, size, and organization appropriate for that application.

OPERATION OVERVIEW

The R3721 constantly monitors the R3051 bus; when a transfer is indicated (by the assertion of the ALE signal from the CPU), the R3721 performs address decoding. Depending on the transfer requested (read or write, and to which address), a number of different events may occur:

The R3722 may perform the transfer itself (between the CPU and one of its internal peripherals);

- It may assert one of its "Chip Select" outputs, indicating that a peripheral under its control is requested for the transfer. In this case, the R3722 will provide the handshake back to the CPU, controlling the wait states required to perform the access.
- It may decode the reference as a reference to the boot prom space. In this case, the R3722 will perform a 32-bit transfer, or a series of 8-bit transfers, from the boot prom, depending on how the system has configured the boot prom interface.
- It may decode the reference as a DRAM reference, in which case it will assert its DRAMSel output, indicating that the external DRAM controller is required to perform the transfer. In this case, the DRAM controller is responsible for wait state generation, allowing the tightest possible coupling (and timing handshake) between DRAM and the CPU.

In addition to handling data transfers, the R3722 also performs basic peripheral functions. For these functions, it uses logic blocks which are operation and software compatible with the 825x series of peripheral devices.

Finally, the R3722 performs DMA transfers between various memory regions. The R3722 is capable of using the R3051 bus at the same transfer speeds as those supported by the processor itself, resulting in the highest levels of DMA performance.

APPLICATION OVERVIEW

The R3722 is designed to interface directly to an R3051 family CPU. Additionally, the R3722 works seamlessly with the R3721 DRAM Controller and R3720 Bus Exchanger. Figure 2 illustrates the construction of an R3051 family systems using this chip set.

6

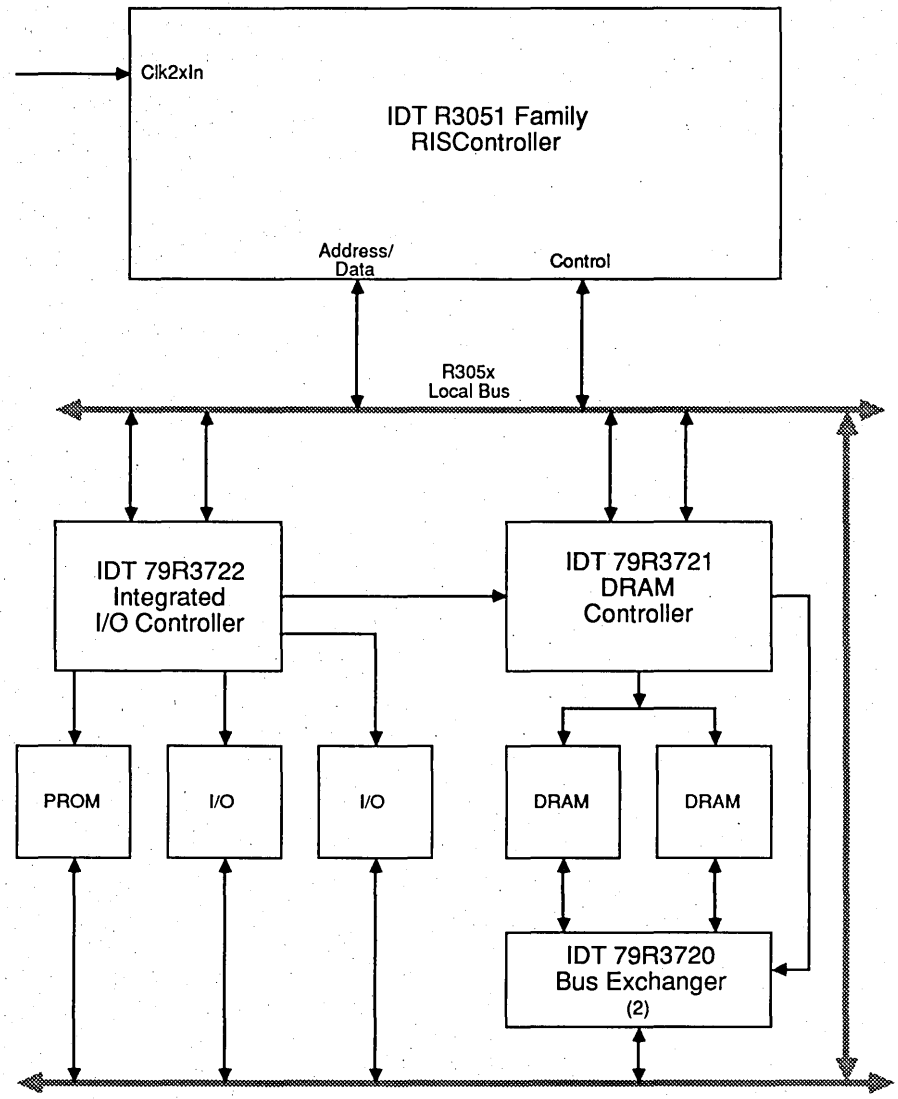


Figure 2. R3051 and RISChipset in an Embedded System

Integrated Device Technology, Inc.

RISC CPU WRITE BUFFER

IDT79R3020

FEATURES:

- Temporary storage buffers to enhance the performance of the IDT79R3000 RISC CPU processor
- Allows for write operations by the RISC CPU processor during Run cycles
- Each Write Buffer has four locations to handle an 8-bit address slice and a 9-bit data slice (including a parity bit)
- High-speed CEMOS™ technology
- Pin, functionally and software compatible with the MIPS Computer Systems R2020 Write Buffer
- Speeds from 16.7 to 40 MHz
- Military product compliant to MIL-STD-883, Class B

DESCRIPTION:

The IDT79R3020 Write Buffer enhances the performance of IDT79R3000 systems by allowing the processor to perform write operations during Run cycles instead of resorting to time-consuming stall cycles. Each IDT79R3020 device handles an 8-bit slice of address, and a 9-bit slice of data (one parity bit per byte); thus, four IDT79R3020s provide 4-deep buffering of 32 bits of address and 36 bits of data and parity. Figure 1 illustrates the functional position of the Write Buffer in an IDT79R3000 system.

Whenever the processor performs a write operation, the Write Buffer captures the output data and its address (including the access type bits). The Write Buffer can hold up to four data–address sets while it waits to pass the data on to main memory. Transfers from the processor to the write buffers occur synchronously at the cycle rate of the processor and the write buffer signals the processor if it is unable to accept data. The write buffer also provides a set of handshake signals to communicate with a main memory controller and coordinate the transfer of write data to main memory.

The sections that follow describe these IDT79R3020 Write Buffer interfaces:

- the processor-Write Buffer interface
- the Write Buffer-main memory interface
- a miscellaneous, Write Buffer-board control interface.

WRITE BUFFER

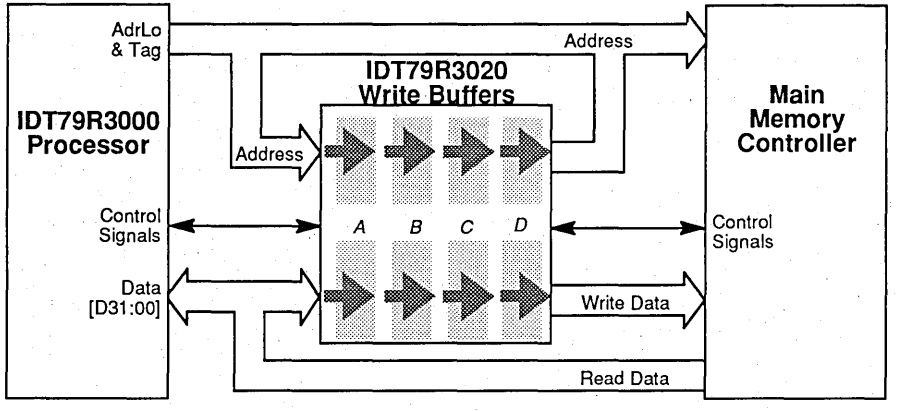


Figure 1. The IDT79R3020 Write Buffer in an IDT79R3000 System

6

CEMOS is a trademark of Integrated Device Technology, Inc.

WRITE BUFFER - IDT79R3000 PROCESSOR INTERFACE

Figure 2 shows the signals comprising the Write Buffer interface to the IDT79R3000 (all descriptions assume that four IDT79R3020 Write Buffers are used to implement a 32-bit, buffered interface). The AdrLo bus and Tag bus bits from the processor are both connected to the Write Buffer to form a 32-bit physical address that is captured by the buffers. Thirty-two bits of data, four bits of parity, and two access type bits are also captured by the Write Buffer. The paragraphs that follow describe the Write Buffer-processor interface signals and the timing of processor-to-Write Buffer data transfers.

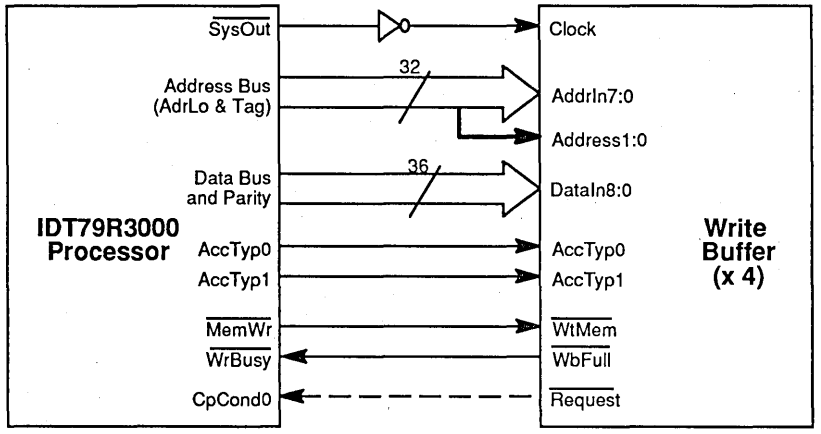


Figure 2. Write Buffer — IDT79R3000 Processor Interface

Write Buffer-Processor Interface Signals

Clock

An inverted version of the IDT79R3000's SysOut signal from the IDT79R3000 processor that synchronizes data transfers. The Write Buffer uses the trailing edge of *Clock* to latch the contents of the AdrLo bus and uses the leading *Clock* edge to latch the contents of the Data and Tag buses.

DataIn8:0

Nine input data lines from the IDT79R3000 processor's Data bus (eight bits of data and one bit of parity).

AddrIn7:0

Eight input address lines from the IDT79R3000 processor. The address lines are taken from the AdrLo and Tag buses.

Address1:0

The two least significant address bits from the IDT79R3000 processor. These two address bits must be connected to all four Write Buffers and are used in conjunction with the access type (*AccTyp1:0*) signals, the *Position1:0* signals, and the *BigEndian* signal to determine which byte(s) in a word are being written into a particular Write Buffer.

AccTypIn1:0

The access type signals from the IDT79R3000 processor specifying the size of a data access: word, tri-byte, half-word, or byte.

WtMem

This input is connected to the MemWr signal from the IDT79R3000 processor that is asserted whenever the processor is performing a store (write) operation.

Request

The primary purpose of this signal is to request access to memory and is described later when the Write Buffer-Main Memory Interface is discussed. The Request signal can also be connected to the CpCond0 input of the IDT79R3000 and can then be tested by software to determine if there is any data in the Write Buffer. Since Request is deasserted if there is no data in the Write Buffer, software can determine if a previous write operation (for example, to an I/O device) has been completed before initiating a read or read status operation from that device.

WbFull

The Write Buffer asserts this signal to the IDT79R3000's WrBusy input whenever it cannot accept any more data; that is, when the current write will fill the buffer or the buffer has all address-data pairs occupied. The IDT79R3000 processor performs a write-busy stall if it needs to store data while the WbFull/WrBusy signal is asserted.

Data & Address Connections

Figure 3 illustrates how four Write Buffers are connected to the address and data outputs of the IDT79R3000 processor.

Address Inputs

Each Write Buffer device has eight address inputs (AdrIn7:0). The four low–order bits (AdrIn3:0) are clocked into the device on the trailing edge of the Clock signal and are taken from the IDT79R3000's AdrLo bus. The four high–order bits (AdrIn7:4) are clocked into the device on the rising edge of the Clock signal and are taken from the IDT79R3000's Tag bus.

Each device also has separate inputs (Address1, Address0) for the two low-order bits from the AdrLo bus. These bits must be input to each device since they comprise the byte pointer. Note in Figure 3 that the two low-order AdrIn inputs (AdrIn1:0) to Write Buffer device 0 are connected to ground since the Address1, Address0 inputs already supply these bits to the device.

Data Inputs

Each Write Buffer device has nine data inputs that are clocked into the device on the leading edge of the Clock signal and are taken from the IDT79R3000's Data bus. In Figure 3, each device captures eight bits of data and one bit of parity. Also note that the data bits assigned to each device correspond to the address bits connected to the device. This arrangement is required since data

selection is dependent on a combination of the AccType signals and the two low order address bits. The arrangement also simplifies system utilization of the "Read Error Address" feature described later.

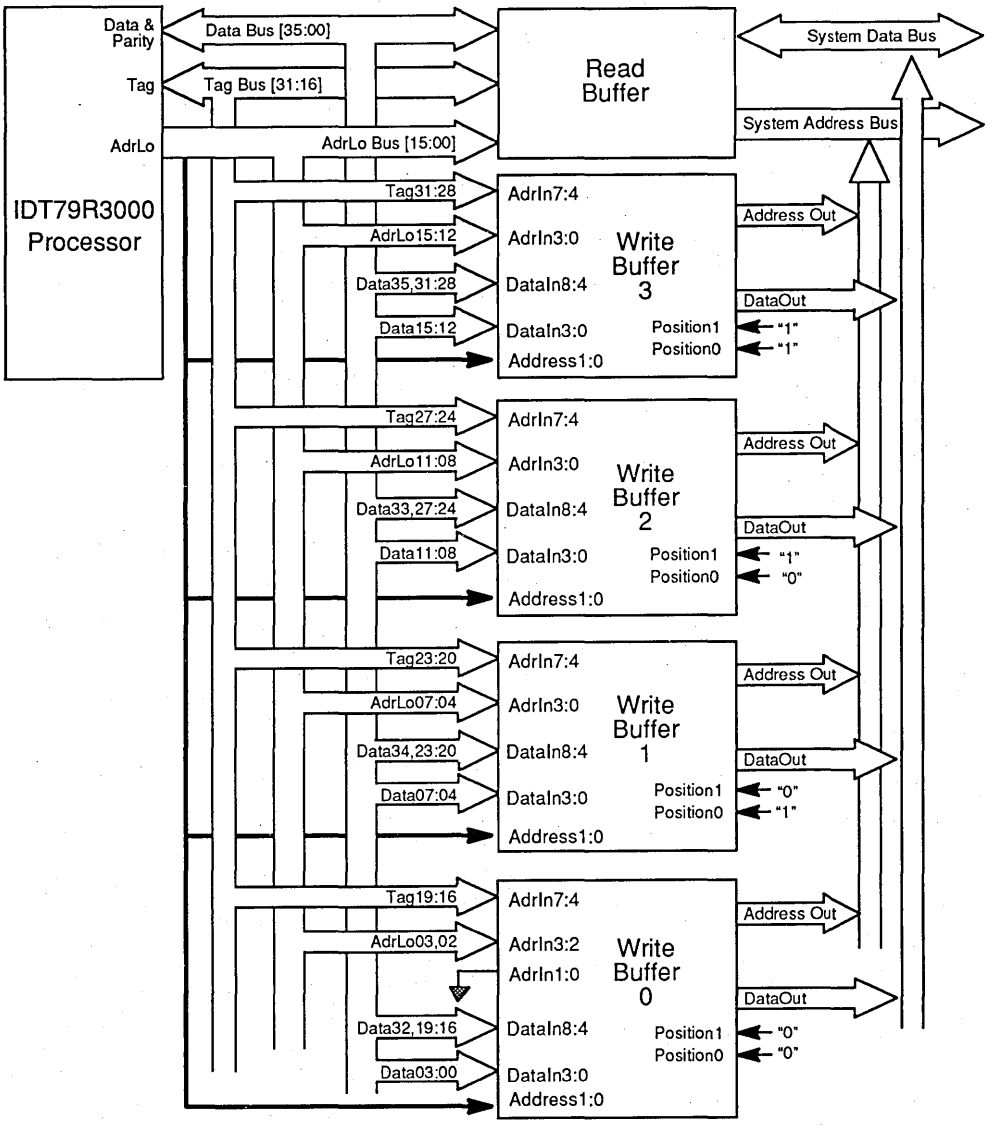


Figure 3. Write Buffer Data and Address Line Connections

The *Position1* and *Position0* signals shown in Figure 3 specify the nibble position within a halfword that each write buffer device comprises.

Write Buffer - Processor Timing

Transfers between the processor and the Write Buffers occur synchronously: the Clock signal from the processor is input to the Write Buffers and used to clock the address and data information into the Write Buffers' latches. Figure 4 illustrates the timing for the processor-Write Buffer interface.

When the WrtMem signal is asserted, the low-order address bits, and the Address 1:0 inputs, are latched on the trailing edge of the Clock signal (1). The rising edge of Clock (2) is used to latch the high-order address bits, the access type inputs and the contents of the data bus.

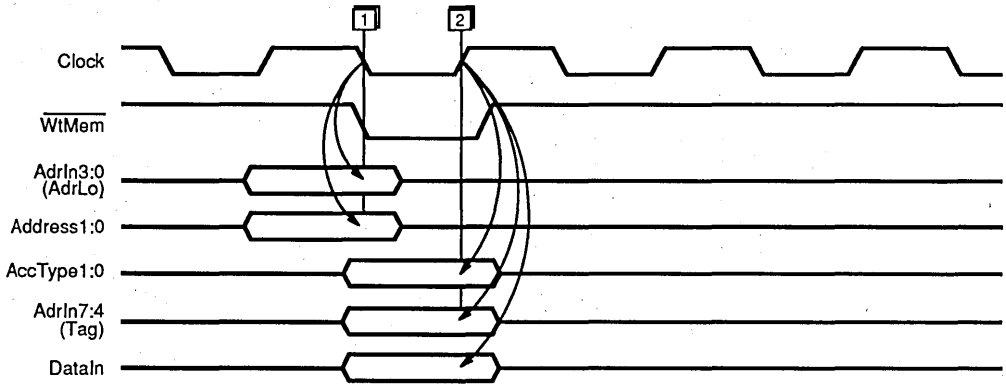


Figure 4. Processor — Write Buffer Interface Timing

WRITE BUFFER - MAIN MEMORY INTERFACE

Figure 5 shows the signals comprising the Write Buffer interface to main memory. This interface is essentially decoupled from the Write Buffer-processor interface: although some synchronization of the memory interface signals and the Clock signal is required, the handshaking signals in this interface have no direct connection to the operation of the Write Buffer-processor interface.

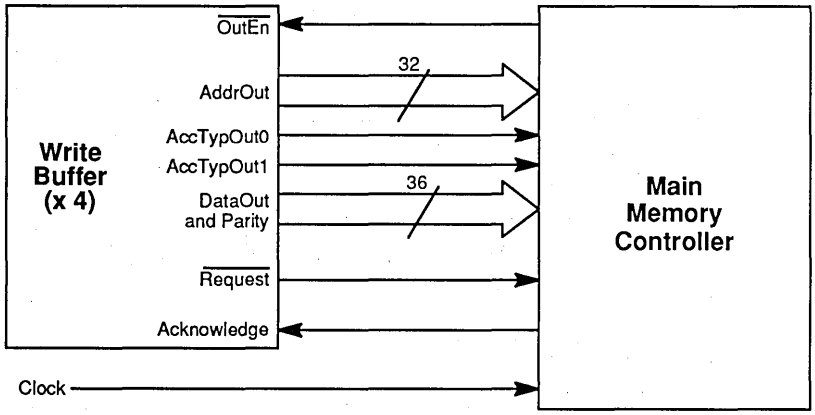


Figure 5. Write Buffer — Main Memory Interface

Write Buffer - Main Memory Interface Signals

Each Write Buffer provides the following signals that comprise the interface to a main memory controller:

AddrOut 7:0
Eight address line output from each Write Buffer.

DataOut 8:0
Nine data lines from each Write Buffer (eight bits of data and one bit of parity).

AccTypOut 1:0
The access type signals from the Write Buffer specifying the size of a data access: word, tri-byte, half-word, or byte.

$\overline{\text{OutEn}}$
The memory controller asserts this write input to enable the tri-state outputs of the IDT79R3020 address and data signals.

$\overline{\text{Request}}$
The Write Buffer asserts this signal to inform the main memory system that it has data to be written to memory.

Acknowledge
The main memory system asserts this signal when it has captured the data presented by the Write Buffer on the DataOut lines.

Write Buffer - Main Memory Interface Timing

Figure 6 illustrates the timing for the transfer of data from the Write Buffer to the main memory system. The sequence illustrated in this figure is as follows:

1) When the Write Buffer has a data-address pair for transfer to the memory system, it asserts the Request signal.
2) When memory system is ready to handle the Write Buffer data, it asserts the OutEn signal to enable the Write Buffers' address and data outputs onto the system buses.
3) When memory system no longer requires the Write Buffer address and data outputs, it asserts the Acknowledge signal. The Write Buffer responds to this signal by discarding the address-data pair that was just output.
4) The memory system can deassert the OutEn signal to return the Write Buffers' address and data outputs to their tri–state condition.
5) Since the Request signal remains asserted, the memory system asserts the OutEn signal again to enable the next address-data pair onto the system buses.
6) When memory system has accepted the second address-data pair, it again asserts the Acknowledge signal. If the Write Buffer is now empty, it responds to this signal by deasserting the Request signal.

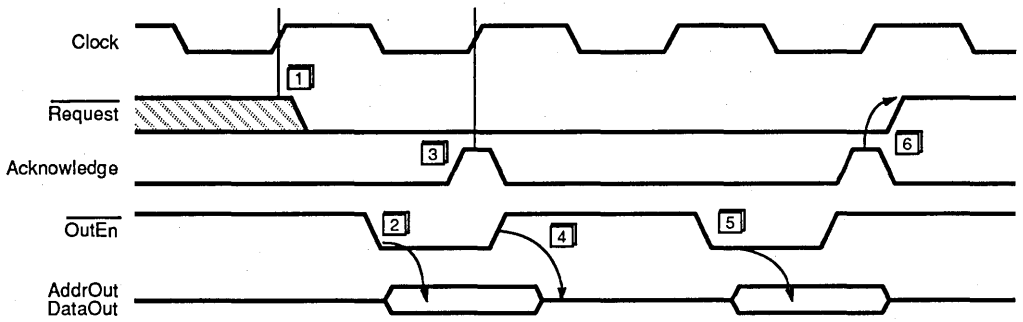


Figure 6. Write Buffer — Main Memory Interface Timing

Note that the buffer's interface to main memory is not completely asynchronous: assertion of the Request signal by the Write Buffer is synchronized with the rising edge of Clock, and the Acknowledge signal input by main memory has a minimum set up and hold time in relation to the Clock signal.

MISCELLANEOUS WRITE BUFFER - BOARD LOGIC INTERFACE

The Write Buffers support several functions that utilize signals that do not fit neatly into the descriptions of either the processor or main memory interfaces. These functions and signals typically involve miscellaneous logic on a CPU board and include the following:

- byte gathering
- configuration connections (Big Endian, Position 1:0)
- address matching logic
- error address latch logic

The sections that follow describe each of these categories.

Byte Gathering

The Write Buffers perform byte (half-word, tri-byte and word) gathering to decrease the number of write transfers to same location; that is, sequential writes to the same WORD address have their data combined into the same address-data pair buffer.

Byte gathering is prohibited in the address-data pair that is currently available to the memory controller. Thus, the first write into an empty Write Buffer will not have subsequent writes gathered into it because it is currently available for output to memory. Writes to the same location (byte) may be overwritten in the Write Buffer if the gathering is not prohibited by the preceding rule.

The Write Buffers present address-data pairs to the main memory controller in the sequence in which they were received from the processor except in the case of gathered data, where bytes or half words can be collected and written to main memory in a single write operation. If the address-data pair buffer is scheduled to be output, then gathering is inhibited and the buffer contents are presented to the main memory controller. Subsequent writes are then placed in another buffer. No reliance should be placed in any aspect of gathering (except that it only involves sequential writes to the same word address) as it is not readily deterministic. Non-sequential writes to the same word address are not gathered.

Note that gathering can require that two main memory controller references be used to empty a single Write Buffer entry. For example, this can occur if Bytes 0 and 3 of a word are sequentially written. Where order in writing is important, such as in I/O controllers, software should avoid sequential accesses to the same word. In cases where write-read access ordering is important but reading of the write location is not desired, such as during I/O, then a write followed by a write to a dummy location followed by a read of the dummy location will insure the first write has occurred before continuing. Alternatively, the Request signal can be tested to determine that the Write Buffer is empty.

Configuration Logic Connections

Because of their byte gathering capability, each buffer device internally maintains a record of each valid byte in an address/data pair. To do this, each device must have a way of determining which data bits within a word it is handling. The following signals determine how the write buffers handle data that is written to the devices:

- **Position 1, Position 0** - these signals (in conjunction with BigEndian) determine how each Write Buffer decodes the Address 1/0 and AccType 1/0 to determine if it should store the data inputs. Refer to Figure 3 for an illustration of how data bits are assigned to Write Buffer devices based on their *position*.
- BigEndian - When asserted, byte 0 is the leftmost, most significant byte (big-endian): when deasserted, byte 0 is the rightmost, least-significant byte (little-endian).
- **Address 1, Address 0** - these signals (taken from the AdrLo bus) must be connected to all buffer devices since they determine which byte within a word is being accessed.

- **AccType 1, AccType 0** - these inputs signals specify the data size of a write operation as shown in Table 1.

Table 1 shows how these signals operate to specify how bytes are saved within the Write Buffers.

Access Type 1 0	Address 1 0	Bytes Accessed: Big–Endian 31 … 0	Bytes Accessed: Little–Endian 31 … 0
1 1 (word)	0 0	0 1 2 3	3 2 1 0
1 0 (triple–byte)	0 0	0 1 2	2 1 0
	0 1	1 2 3	3 2 1
0 1 (halfword)	0 0	0 1	1 0
	1 0	2 3	3 2
0 0 (byte)	0 0	0	0
	0 1	1	1
	1 0	2	2
	1 1	3	3

Table 1. Byte Specifications for Write Operations

The lower two address bits of the device in position zero (as determined by the two POSITION inputs) are inhibited; that is, they are not stored directly as they are output on the AdrLo bus. Instead, on output, the lower two address bits are generated from the indication of the positions of the valid data bytes as determined by above table.

MatchOut/MatchIn Logic and Read Conflicts

Whenever the processor references main memory (either a write or a read reference), the Write Buffers compare the word address from the CPU with the word addresses stored in the buffers. If any word address matches, the buffers assert signals that can be used by the main memory controller to ensure that the Write Buffer is emptied before the read access with the conflicting address has been performed.

Figure 7 illustrates the Write Buffer signals involved in address comparison logic. Each write buffer provides four output signals (MatchOut A, B, C, and D) which correspond to the four buffer ranks (A, B, C, D) in each device as shown in Figure 1. These MatchOut signals can be externally NANDed as shown in Figure 7 to determine if the address being input matches those in any rank of the Write Buffer.

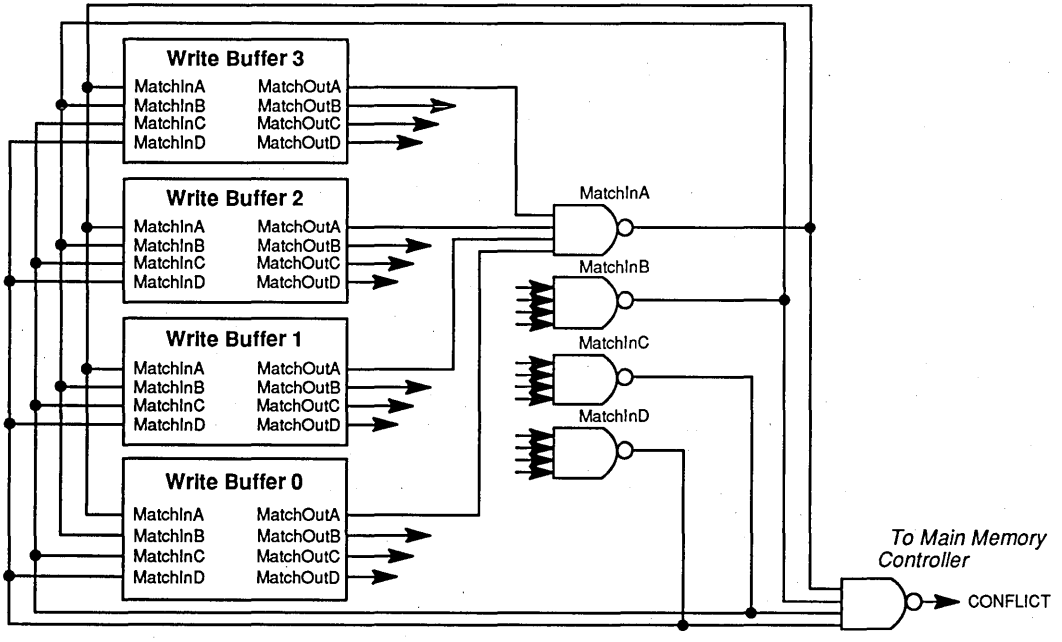


Figure 7. Write Buffer MatchOut/MatchIn Logic

The outputs of the NAND gates are fed into Write Buffers via the MatchIn A, B, C, and D signals and are used within each device as part of the byte gathering logic. The NAND gate outputs can be NANDed together as shown in Figure 7 with the resultant signal used (in conjunction with the processor's $\overline{\text{MEMRD}}$ signal) to alert the main memory controller logic that there is a pending buffered write that conflicts with a just-issued read. The main memory controller can then delay the read access until the Request signal is deasserted indicating that the Write Buffer has been emptied.

Error Address Latch

The write buffer incorporates an internal latch that can be loaded with one of the buffered addresses and subsequently enabled out onto the data lines. This feature can be used by error handling routines to read an address back from the Write Buffer and analyze or recover from certain bus errors. Figure 8 shows the signals involved in operation of this latch.

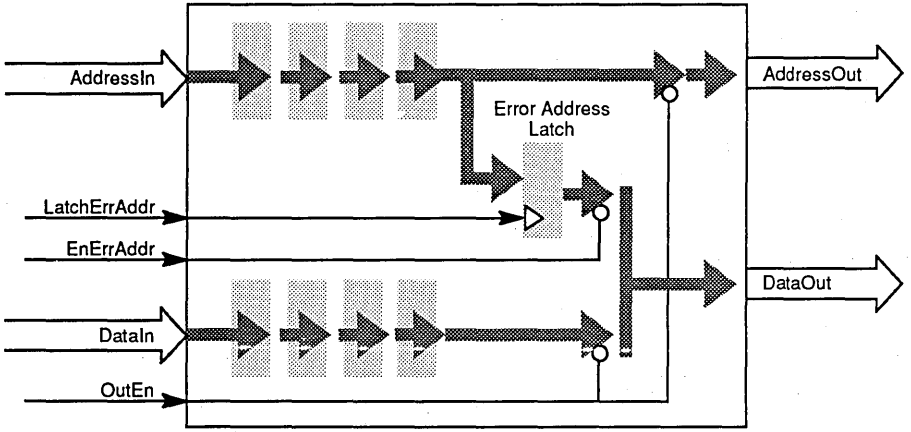


Figure 8. The Write Buffer Error Address Latch

When the LatchErrAddr signal is asserted, the address currently available to the address outputs of the Write Buffer is latched into the internal latch. This address can then be output on the DataOut lines by asserting the EnErrAdr signal so that the processor can read the address in as data. Refer to the AC specifications for timing parameters of the signals associated with the error address latch.

6

ABSOLUTE MAXIMUM RATINGS[1, 3]

SYMBOL	RATING	COMMERCIAL	MILITARY	UNIT
V_{TERM}	Terminal Voltage with Respect to GND	–0.5 to +7.0	–0.5 to +7.0	V
T_A	Operating Temperature	0 to +70	–55 to +125	°C
T_{BIAS}	Temperature Under Bias	–55 to +125	–65 to +135	°C
T_{STG}	Storage Temperature[2]	–55 to +125	–65 to +150	°C
V_{IN}	Input Voltage	–0.5 to +7.0	–0.5 to +7.0	V

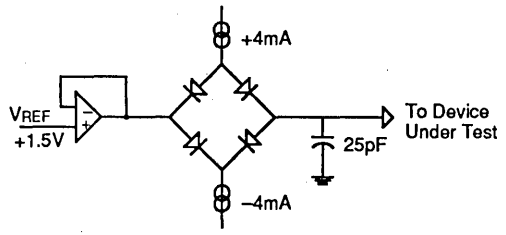
NOTES:

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. V_{IN} minimum = –3.0V for pulse width less than 15ns. V_{IN} should not exceed V_{CC} +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

GRADE	AMBIENT TEMPERATURE	GND	Vcc
Military	–55°C to +125°C	0V	5.0 ± 10%
Commercial	0°C to +70°C	0V	5.0 ± 5%

OUTPUT LOADING FOR AC TESTING



DC ELECTRICAL CHARACTERISTICS — COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0 V ± 5%)

SYMBOL	PARAMETER	TEST CONDITIONS	16.67 MHz MIN.	16.67 MHz MAX.	20.0 MHz MIN.	20.0 MHz MAX.	25.0 MHz MIN.	25.0 MHz MAX.	33.33MHz MIN.	33.33MHz MAX.	40 MHz MIN.	40 MHz MAX.	UNIT
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = –4mA	3.5	—	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	—	2.4	V
V_{IH}	Input HIGH Voltage[1]		2.4	—	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{IL}	Input LOW Voltage[2]		—	0.8	—	0.8	—	0.8	—	0.8	—	0.8	V
C_{IN}	Input Capacitance		10	—	10	—	10	—	10	—	10	—	pF
C_{OUT}	Output Capacitance		10	—	10	—	10	—	10	—	10	—	pF
I_{CC}	Operating Current	V_{CC} = Max	—	50	—	60	—	70	—	80	—	90	mA
I_{IH}	Input HIGH Leakage	V_{IH} = V_{CC}	—	10	—	10	—	10	—	10	—	10	µA
I_{IL}	Input LOW Leakage	V_{IL} = GND	–10	—	–10	—	–10	—	–10	—	–10	—	µA
I_{OZ}	Output Tri–state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–40	40	–40	40	–40	40	–40	40	–40	40	µA

DC ELECTRICAL CHARACTERISTICS — MILITARY TEMPERATURE RANGE (T_A = –55°C to +125°C, V_{CC} = +5.0 V ± 10%)

SYMBOL	PARAMETER	TEST CONDITIONS	16.67 MHz MIN.	16.67 MHz MAX.	20.0 MHz MIN.	20.0 MHz MAX.	25.0 MHz MIN.	25.0 MHz MAX.	33.0 MHz MIN.	33.0 MHz MAX.	UNIT
V_{OH}	Output HIGH Voltage	V_{CC} = Min, I_{OH} = –4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min, I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{IH}	Input HIGH Voltage[1]		2.4	—	2.4	—	2.4	—	2.4	—	V
V_{IL}	Input LOW Voltage[2]		—	0.8	—	0.8	—	0.8	—	0.8	V
C_{IN}	Input Capacitance		10	—	10	—	10	—	10	—	pF
C_{OUT}	Output Capacitance		10	—	10	—	10	—	10	—	pF
I_{CC}	Operating Current	V_{CC} = Max	—	50	—	60	—	70	—	80	mA
I_{IH}	Input HIGH Leakage	V_{IH} = V_{CC}	—	10	—	10	—	10	—	10	µA
I_{IL}	Input LOW Leakage	V_{IL} = GND	–10	—	–10	—	–10	—	–10	—	µA
I_{OZ}	Output Tri–state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–40	40	–40	40	–40	40	–40	40	µA

NOTES:

1. V_{IH} should be held above V_{CC} + 0.5 Volts.
2. V_{IL} Min. = –3.0V for pulse width less than 15ns. V_{IL} should not fall below –0.5 Volts for longer periods.

AC ELECTRICAL CHARACTERISTICS — COMMERCIAL TEMPERATURE RANGE (T_A + 0°C to 70°C, V_{CC} = +5.0V ± 5%)

SYMBOL	PARAMETER	16.67 MHz MIN.	16.67 MHz MAX.	20.0 MHz MIN.	20.0 MHz MAX.	25.0 MHz MIN.	25.0 MHz MAX.	33.33 MHz MIN.	33.33 MHz MAX.	40.0 MHz MIN.	40.0 MHz MAX.	UNIT
t1	Addrin (3:0) to Clock falling setup	8	—	7	—	6	—	3	—	3	—	ns
t2	Addrin (3:0) from Clock falling hold	4	—	4	—	4	—	3	—	3	—	ns
t3	Address 1:0 to Clock falling setup	8	—	7	—	6	—	3	—	3	—	ns
t4	Address 1:0 from Clock falling hold	4	—	4	—	4	—	3	—	3	—	ns
t5	Access Type 1:0 to Clock rising setup	7	—	6	—	5	—	4	—	4	—	ns
t6	Access Type 1:0 from Clock rising hold	3	—	3	—	2	—	2	—	2	—	ns
t7	Addrin (7:4) to Clock rising setup	7	—	5	—	4	—	4	—	4	—	ns
t8	Addrin (7:4) from Clock rising hold	3	—	3	—	2	—	1	—	1	—	ns
t9	DataIn (8:0) to Clock rising setup	7	—	5	—	4	—	4	—	4	—	ns
t10	DataIn (8:0) from Clock rising hold	3	—	3	—	2	—	1	—	1	—	ns
t11	$\overline{\text{WrtMem}}$ to Clock rising setup	10	—	8	—	7	—	6	—	6	—	ns
t12	$\overline{\text{WrtMem}}$ from Clock rising hold	6	—	5	—	4	—	3	—	3	—	ns
t13	Request from Clock rising	—	32	—	30	—	22	—	16	—	16	ns
t14	Acknowledge to Clock rising setup	12	—	11	—	6	—	4	—	4	—	ns
t15	Acknowledge from Clock rising hold	7	—	6	—	5	—	3	—	3	—	ns
t16	LatchErrAdr to Acknowledge rising	5	—	5	—	5	—	3	—	3	—	ns
t17	$\overline{\text{WbFull}}$ active from Clock rising	—	21	—	19	—	17	—	9	—	9	ns
t18	$\overline{\text{WbFull}}$ inactive from Clock rising	—	21	—	19	—	11	—	9	—	9	ns
t19	OutEn to AddrOut (7:0), DataOut (8:0) valid	2	15	2	15	2	15	2	12	2	12	ns
t20	OutEn to AddrOut (7:0), DataOut (8:0) tri-state	2	15	2	15	2	15	2	12	2	12	ns
t21	MatchOut (ABCD) from Clock rising	—	24	—	22	—	20	—	15	—	15	ns
t22	MatchIn (ABCD) to Clock rising setup	10	—	9	—	8	—	5	—	5	—	ns
t23	MatchIn (ABCD) from Clock rising hold	3	—	3	—	3	—	3	—	3	—	ns
t24	$\overline{\text{EnErrAdr}}$ to Data (error latch) valid	2	15	2	15	2	15	2	15	2	15	ns
t25	$\overline{\text{EnErrAdr}}$ to Data (error latch) tri-state	2	15	2	15	2	15	2	15	2	15	ns
t26	Address/Data out from Clock rising	—	30	—	27	—	24	—	16	—	16	ns
t27	$\overline{\text{Reset}}$ to Clock rising, set-up	10	—	10	—	10	—	8	—	8	—	ns
t28	$\overline{\text{Reset}}$ from Clock rising, hold	3	—	2	—	1	—	1	—	1	—	ns
t29	Reset low pulse width	8	—	8	—	8	—	8	—	8	—	cycles
t30	$\overline{\text{WbFull}}$ High from Clock rising (after $\overline{\text{Reset}}$)	—	22	—	21	—	20	—	11	—	11	ns
t31	$\overline{\text{Request}}$ High from $\overline{\text{Reset}}$ low	—	20	—	19	—	18	—	16	—	16	ns
t32	Access TypOut 1:0 low from $\overline{\text{Reset}}$ low	—	28	—	26	—	25	—	23	—	23	ns
t33	Match Out (ABCD) Low from $\overline{\text{Reset}}$ low	—	21	—	20	—	20	—	15	—	15	ns
t34	Address/Data out tri-state from $\overline{\text{Reset}}$ low (OutEn negated)	—	32	—	30	—	27	—	23	—	23	ns
t35	Access TypeOut from Clock rising	—	32	—	30	—	27	—	23	—	23	ns
tcyc	Clock Pulse Width	60	2000	50	2000	40	2000	30	2000	25	2000	ns
tckhigh	Clock High Pulse Width	24	—	20	—	16	—	12	—	10	—	ns
tcklow	Clock Low Pulse Width	24	—	20	—	16	—	12	—	10	—	ns

AC ELECTRICAL CHARACTERISTICS — MILITARY TEMPERATURE RANGE (T_A + −55°C to 125°C, V_{CC} = +5.0V ± 10%)

SYMBOL	PARAMETER	16.67 MHz MIN.	16.67 MHz MAX.	20.0 MHz MIN.	20.0 MHz MAX.	25.0 MHz MIN.	25.0 MHz MAX.	33.0 MHz MIN.	33.0 MHz MAX.	UNIT
t1	Addrin (3:0) to Clock falling setup	8	—	7	—	6	—	3	—	ns
t2	Addrin (3:0) from Clock falling hold	4	—	4	—	4	—	3	—	ns
t3	Address 1:0 to Clock falling setup	8	—	7	—	6	—	3	—	ns
t4	Address 1:0 from Clock falling hold	4	—	4	—	4	—	3	—	ns
t5	Access Type 1:0 to Clock rising setup	7	—	6	—	5	—	4	—	ns
t6	Access Type 1:0 from Clock rising hold	3	—	3	—	2	—	2	—	ns
t7	Addrin (7:4) to Clock rising setup	7	—	5	—	4	—	4	—	ns
t8	Addrin (7:4) from Clock rising hold	3	—	3	—	2	—	1	—	ns
t9	DataIn (8:0) to Clock rising setup	7	—	5	—	4	—	4	—	ns
t10	DataIn (8:0) from Clock rising hold	3	—	3	—	2	—	1	—	ns
t11	$\overline{\text{WrtMem}}$ to Clock rising setup	10	—	8	—	7	—	6	—	ns
t12	$\overline{\text{WrtMem}}$ from Clock rising hold	6	—	5	—	4	—	3	—	ns
t13	Request from Clock rising	—	32	—	30	—	22	—	16	ns
t14	Acknowledge to Clock rising setup	12	—	11	—	6	—	4	—	ns
t15	Acknowledge from Clock rising hold	7	—	6	—	5	—	3	—	ns
t16	LatchErrAdr to Acknowledge rising	5	—	5	—	5	—	3	—	ns
t17	$\overline{\text{WbFull}}$ active from Clock rising	—	21	—	19	—	17	—	9	ns
t18	$\overline{\text{WbFull}}$ inactive from Clock rising	—	21	—	19	—	11	—	9	ns
t19	OutEn to AddrOut (7:0), DataOut (8:0) valid	2	15	2	15	2	15	2	12	ns
t20	OutEn to AddrOut (7:0), DataOut (8:0) tri-state	2	15	2	15	2	15	2	12	ns
t21	MatchOut (ABCD) from Clock rising	—	24	—	22	—	20	—	15	ns
t22	MatchIn (ABCD) to Clock rising setup	10	—	9	—	8	—	5	—	ns
t23	MatchIn (ABCD) from Clock rising hold	3	—	3	—	3	—	3	—	ns
t24	$\overline{\text{EnErrAdr}}$ to Data (error latch) valid	2	15	2	15	2	15	2	15	ns
t25	$\overline{\text{EnErrAdr}}$ to Data (error latch) tri-state	2	15	2	15	2	15	2	15	ns
t26	Address/Data out from Clock rising	—	30	—	27	—	24	—	16	ns
t27	$\overline{\text{Reset}}$ to Clock rising, set-up	10	—	10	—	10	—	8	—	ns
t28	$\overline{\text{Reset}}$ from Clock rising, hold	3	—	2	—	1	—	1	—	ns
t29	$\overline{\text{Reset}}$ low pulse width	8	—	8	—	8	—	8	—	cycles
t30	$\overline{\text{WbFull}}$ High from Clock rising (after $\overline{\text{Reset}}$)	—	22	—	21	—	20	—	11	ns
t31	$\overline{\text{Request}}$ High from $\overline{\text{Reset}}$ low	—	20	—	19	—	18	—	16	ns
t32	Access TypOut 1:0 low from $\overline{\text{Reset}}$ low	—	28	—	26	—	25	—	23	ns
t33	Match Out (ABCD) Low from $\overline{\text{Reset}}$ low	—	21	—	20	—	20	—	15	ns
t34	Address/Data out tri–state from $\overline{\text{Reset}}$ low (OutEn negated)	—	32	—	30	—	27	—	23	ns
t35	Access TypeOut from Clock rising	—	32	—	30	—	27	—	23	ns
tcyc	Clock Pulse Width	60	2000	50	2000	40	2000	30	2000	ns
tckhigh	Clock High Pulse Width	24	—	20	—	16	—	12	—	ns
tcklow	Clock Low Pulse Width	24	—	20	—	16	—	12	—	ns

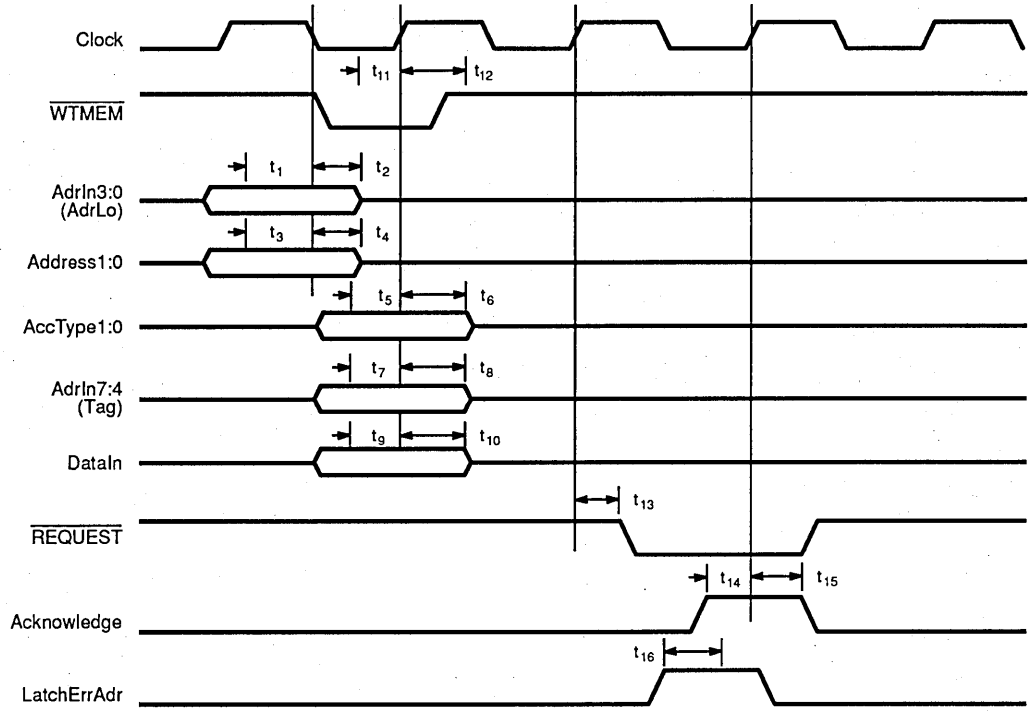


Figure 9. Write Buffer Timing Specifications

6

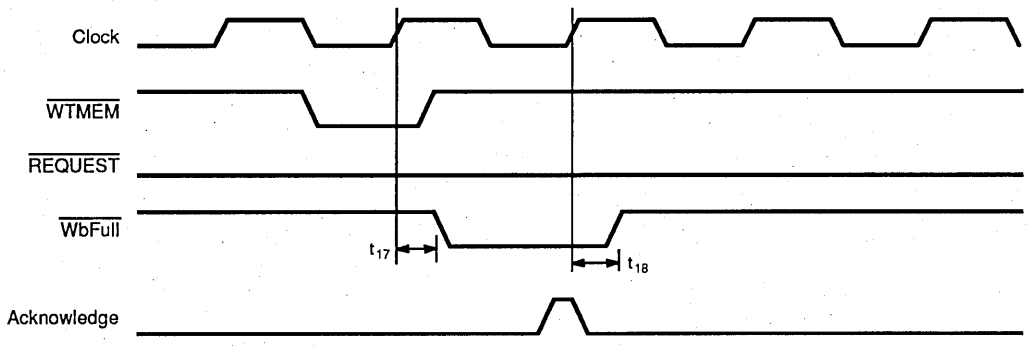


Figure 10. WBFULL Signal Timing Specifications

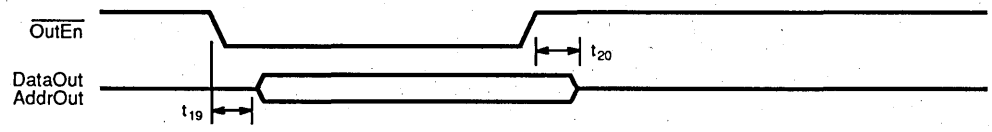


Figure 11. $\overline{\text{OUTEN}}$ Timing Specifications

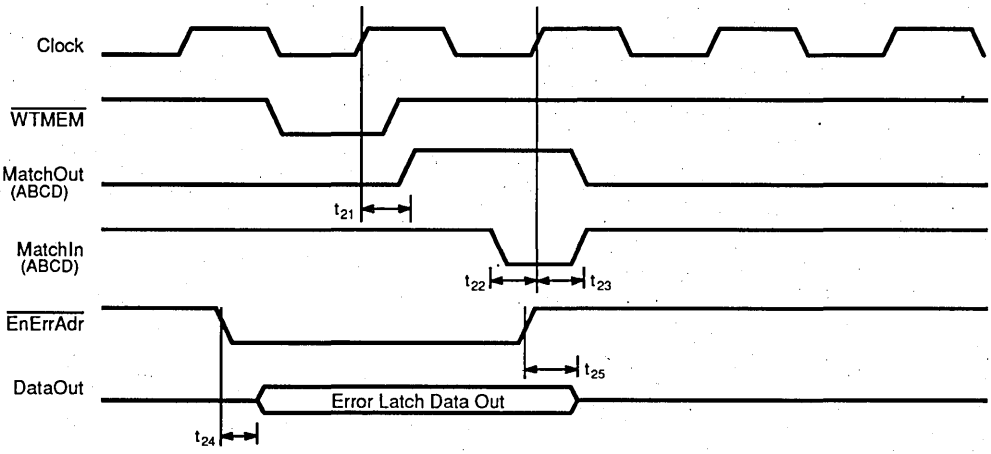


Figure 12. Match and Error Latch Timing Specifications

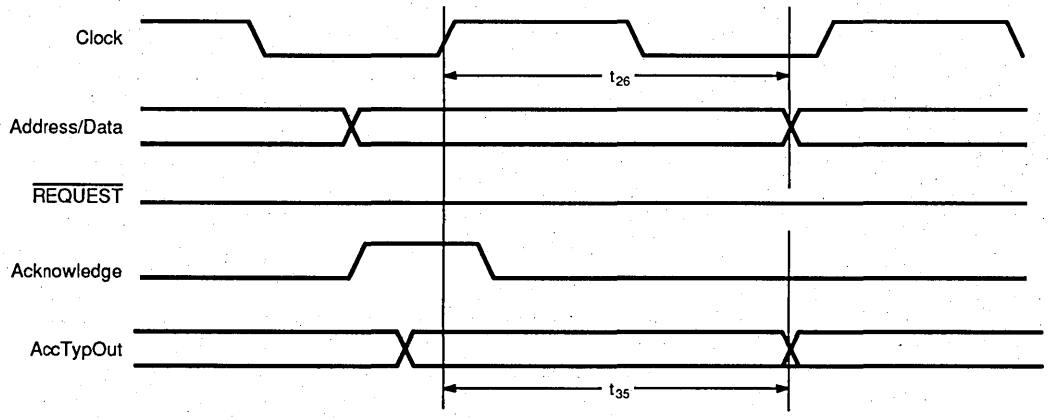


Figure 13. Address/Data Out, Access Type Out

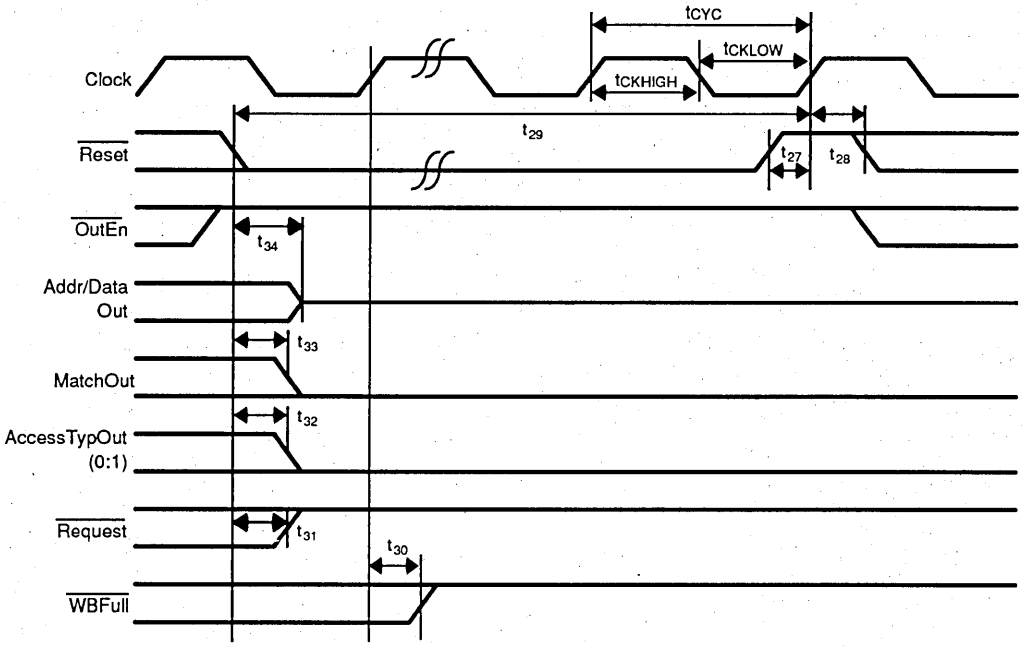
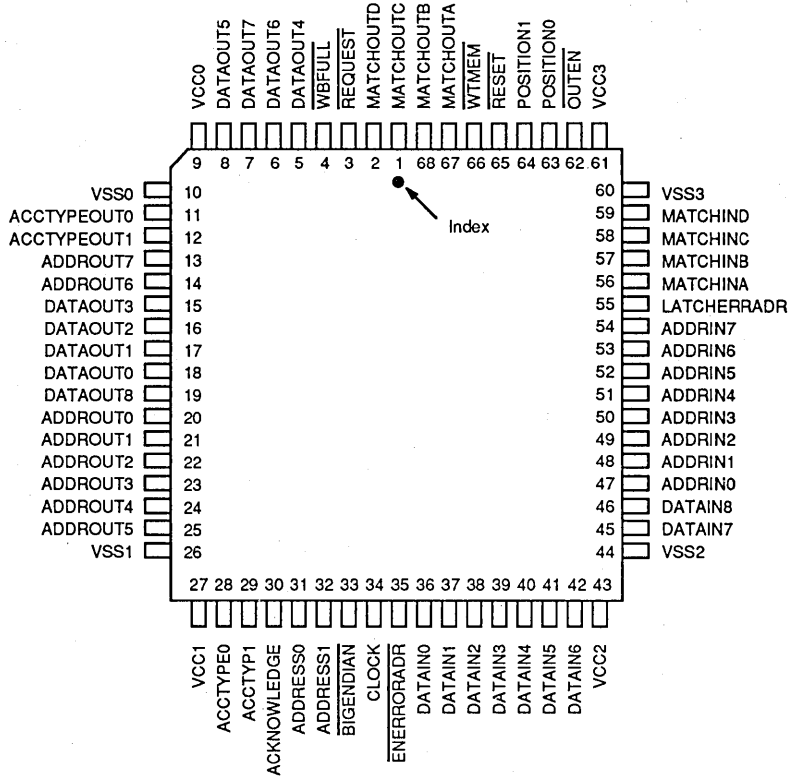


Figure 14. Reset Timing

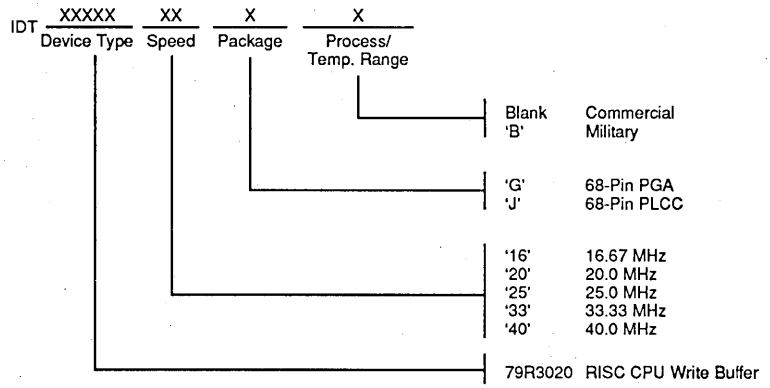
68-PIN CPGA FOR R3020
PIN GRID ARRAY (CERAMIC) – BOTTOM VIEW

	1	2	3	4	5	6	7	8	9	10	11
L		ACC-TYPE0	AC-KNOWL-EDGE	AD-DRESS1	CLOCK	DATA-IN0	DATA-IN2	DATA-IN4	DATA-IN6	VCC2	
K	GND1	VCC1	ACC-TYPE1	AD-DRESS0	$\overline{\text{BIG-ENDIAN}}$	$\overline{\text{EN-ERROR-ADR}}$	DATA-IN1	DATA-IN3	DATA-IN5	GND2	DATA-IN7
J	ADDR-OUT5	ADDR-OUT4								DATA-IN8	ADDR-IN0
H	ADDR-OUT3	ADDR-OUT2								ADDR-IN1	ADDR-IN2
G	ADDR-OUT1	ADDR-OUT0								ADDR-IN3	ADDR-IN4
F	DATA-OUT8	DATA-OUT0								ADDR-IN5	ADDR-IN6
E	DATA-OUT1	DATA-OUT2								ADDR-IN7	LATCH-ERR-ADR
D	DATA-OUT3	ADDR-OUT6								MATCH-INA	MATCH-INB
C	ADDR-OUT7	ACC-TYPE OUT1								MATCH-INC	MATCH-IND
B	ACC-TYPE OUT0	GND0	DATA-OUT7	DATA-OUT4	$\overline{\text{RE-QUEST}}$	MATCH-OUTC	MATCH-OUTA	$\overline{\text{RESET}}$	PO-SITION0	VCC3	GND3
A		VCC0	DATA-OUT5	DATA-OUT6	$\overline{\text{WBFULL}}$	MATCH-OUTD	MATCH-OUTB	$\overline{\text{WTMEM}}$	PO-SITION1	$\overline{\text{OUTEN}}$	

PIN CONFIGURATION
PLASTIC LEADED CHIP CARRIER
(TOP VIEW)



ORDERING INFORMATION



6

Integrated Device Technology, Inc.

16-BIT CMOS MULTILEVEL PIPELINE REGISTERS

IDT73200
IDT73201

FEATURES:

- IDT73200: Eight 16-bit high-speed pipeline registers
- IDT73201: Seven 16-bit high-speed pipeline registers plus a direct feed-through path
- 12ns to 20ns access time
- Programmable multilevel register configurations
- Powerful instruction set: transfer, hold, load directly
- Functionally replaces four Am29520s
- Read/Write buffer for 32-bit RISC/CISC microprocessors
- Applications as temporary address storage or programmable pipeline registers for DSP products
- Coefficient storage for FIR filters
- Three-state outputs
- TTL-compatible
- Produced with advanced submicron CEMOS™ high-performance technology
- Available in 48-pin plastic and ceramic DIP and 52-pin surface mount PLCC
- Military product compliant to MIL-STD-883, Class B

DESCRIPTION:

The IDT73200 and IDT73201 are mutilevel pipeline registers. With IDT's high-performance CEMOS™ technology, the IDT73200 and IDT73201 have access times of 12ns.

The IDT73200 contains eight 16-bit registers which can be configured as one 8-level, two 4-level , four 2-level or eight 1-level pipeline registers.

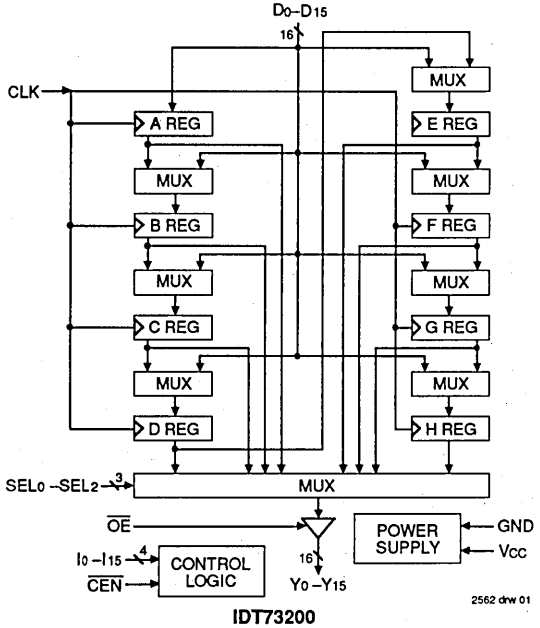
The IDT73201 contains seven 16-bit registers and a direct feed-through path. The seven registers can be configured as one 7-level, a 4-level plus a 3-level, three 2-level or seven 1-level pipeline registers.

An eight-to-one output multiplexer allows data to be read from any one of the registers or from the feed-through path on the IDT73201. Three input control pins (SEL_0–SEL_2) select which of the multiplexer inputs are directed to the output (Y_0–Y_{15}).

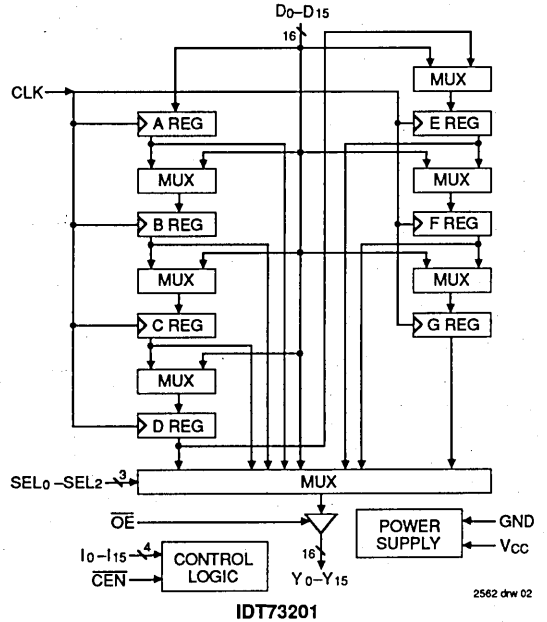
These pipeline registers are ideal for high throughput, vector-oriented operations such as those in digital signal processing (DSP). The IDT73200 and IDT73201 can also be used as quick access scratch pad registers for general purpose computing.

The two pipeline registers are packaged in 48-pin plastic and ceramic DIPs for through–hole designs as well as 52-pin PLCC and LCC for surface mount designs. Military grade product is manufactured in compliance with the latest revision of MIL-STD-883, Class B.

FUNCTIONAL BLOCK DIAGRAMS



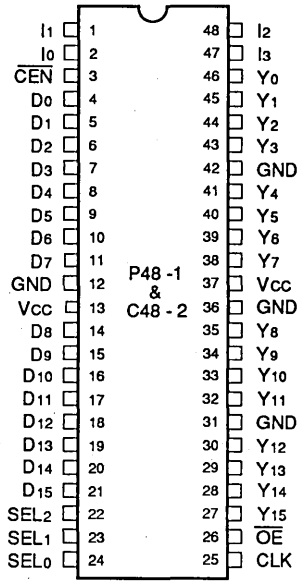
IDT73200



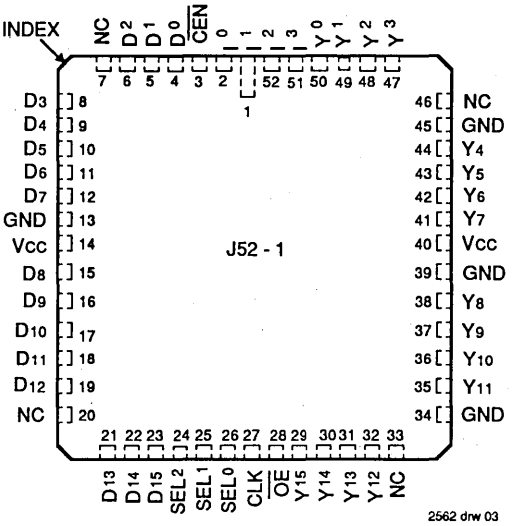
IDT73201

CEMOS is a trademark of Integrated Device Technology Inc.

PIN CONFIGURATIONS



DIP
TOP VIEW



PLCC
TOP VIEW

PIN DESCRIPTIONS

Pin Name	I/O	Description
$D_0 - D_{15}$	I	Sixteen-bit data input port.
$Y_0 - Y_{15}$	O	Sixteen-bit data output port.
$I_0 - I_3$	I	Four control pins to select the register operation performed.
$SEL_0 - SEL_2$	I	Three control pins to select the register appearing at the output.
CLK	I	Clock input.
$\overline{CEN}$	I	Clock enable control pin. When this pin is low, the instruction I_0–I_3 is performed on the registers. When high, no register operation occurs.
$\overline{OE}$	I	Output enable control pin. When this pin is high, the output port Y is in a high impedance state. When low, the output port Y is active.
V_{CC}		Power supply pin, 5V.
GND		Ground pins, 0V.

2562 tbl 01

6

IDT73200 OUTPUT SELECTION

SEL_2	SEL_1	SEL_0	Y Output
0	0	0	$A \rightarrow Y_0 - Y_{15}$
0	0	1	$B \rightarrow Y_0 - Y_{15}$
0	1	0	$C \rightarrow Y_0 - Y_{15}$
0	1	1	$D \rightarrow Y_0 - Y_{15}$
1	0	0	$E \rightarrow Y_0 - Y_{15}$
1	0	1	$F \rightarrow Y_0 - Y_{15}$
1	1	0	$G \rightarrow Y_0 - Y_{15}$
1	1	1	$H \rightarrow Y_0 - Y_{15}$

2562 tbl 02

IDT73201 OUTPUT SELECTION

SEL_2	SEL_1	SEL_0	Y Output
0	0	0	$A \rightarrow Y_0 - Y_{15}$
0	0	1	$B \rightarrow Y_0 - Y_{15}$
0	1	0	$C \rightarrow Y_0 - Y_{15}$
0	1	1	$D \rightarrow Y_0 - Y_{15}$
1	0	0	$E \rightarrow Y_0 - Y_{15}$
1	0	1	$F \rightarrow Y_0 - Y_{15}$
1	1	0	$G \rightarrow Y_0 - Y_{15}$
1	1	1	$D_0 - D_{15} \rightarrow Y_0 - Y_{15}$

2562 tbl 03

IDT73200 INSTRUCTION TABLE

I_3	I_2	I_1	I_0	Mnemonic	Function	Pipeline Levels
0	0	0	0	LDA	$D_0 - D_{15} \rightarrow A$	1
0	0	0	1	LDB	$D_0 - D_{15} \rightarrow B$	1
0	0	1	0	LDC	$D_0 - D_{15} \rightarrow C$	1
0	0	1	1	LDD	$D_0 - D_{15} \rightarrow D$	1
0	1	0	0	LDE	$D_0 - D_{15} \rightarrow E$	1
0	1	0	1	LDF	$D_0 - D_{15} \rightarrow F$	1
0	1	1	0	LDG	$D_0 - D_{15} \rightarrow G$	1
0	1	1	1	LDH	$D_0 - D_{15} \rightarrow H$	1
1	0	0	0	LSHAH	$D_0 - D_{15} \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H$	8
1	0	0	1	LSHAD	$D_0 - D_{15} \rightarrow A \rightarrow B \rightarrow C \rightarrow D$	4
1	0	1	0	LSHEH	$D_0 - D_{15} \rightarrow E \rightarrow F \rightarrow G \rightarrow H$	4
1	0	1	1	LSHAB	$D_0 - D_{15} \rightarrow A \rightarrow B$	2
1	1	0	0	LSHCD	$D_0 - D_{15} \rightarrow C \rightarrow D$	2
1	1	0	1	LSHEF	$D_0 - D_{15} \rightarrow E \rightarrow F$	2
1	1	1	0	LSHGH	$D_0 - D_{15} \rightarrow G \rightarrow H$	2
1	1	1	1	HOLD	Hold All Registers	—

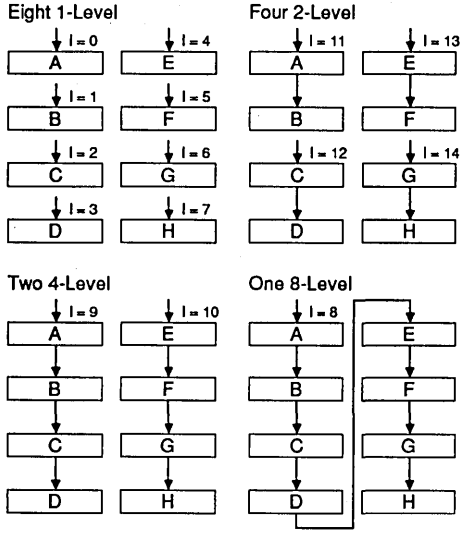
2562 tbl 04

IDT73201 INSTRUCTION TABLE

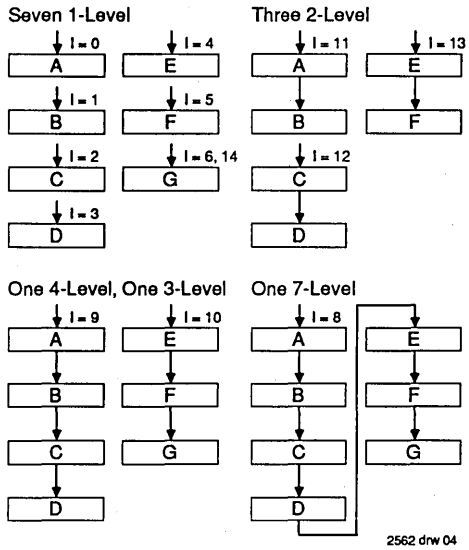
I_3	I_2	I_1	I_0	Mnemonic	Function	Pipeline Levels
0	0	0	0	LDA	$D_0 - D_{15} \rightarrow A$	1
0	0	0	1	LDB	$D_0 - D_{15} \rightarrow B$	1
0	0	1	0	LDC	$D_0 - D_{15} \rightarrow C$	1
0	0	1	1	LDD	$D_0 - D_{15} \rightarrow D$	1
0	1	0	0	LDE	$D_0 - D_{15} \rightarrow E$	1
0	1	0	1	LDF	$D_0 - D_{15} \rightarrow F$	1
0	1	1	0	LDG	$D_0 - D_{15} \rightarrow G$	1
0	1	1	1	HOLD	Hold All Registers	—
1	0	0	0	LSHAG	$D_0 - D_{15} \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$	7
1	0	0	1	LSHAD	$D_0 - D_{15} \rightarrow A \rightarrow B \rightarrow C \rightarrow D$	4
1	0	1	0	LSHEG	$D_0 - D_{15} \rightarrow E \rightarrow F \rightarrow G$	3
1	0	1	1	LSHAB	$D_0 - D_{15} \rightarrow A \rightarrow B$	2
1	1	0	0	LSHCD	$D_0 - D_{15} \rightarrow C \rightarrow D$	2
1	1	0	1	LSHEF	$D_0 - D_{15} \rightarrow E \rightarrow F$	2
1	1	1	0	LDG	$D_0 - D_{15} \rightarrow G$	1
1	1	1	1	HOLD	Hold All Registers	—

2562 tbl 05

IDT73200 PIPELINE CONFIGURATIONS



IDT73201 PIPELINE CONFIGURATIONS



ABSOLUTE MAXIMUM RATINGS(1)

Symbol	Rating	Commercial	Military	Unit
VCC	Power Supply Voltage	-0.5 to +7.0	-0.5 to +7.0	V
VTERM	Terminal Voltage with Respect to GND	–0.5 to VCC + 0.5	–0.5 to VCC + 0.5	V
TA	Operating Temperature	0 to +70	–55 to +125	°C
TBIAS	Temperature Under Bias	–55 to +125	–65 to +135	°C
TSTG	Storage Temperature	–55 to +125	–65 to +155	°C
IOUT	DC Output Current	50	50	mA

NOTE: 2562 tbl 06

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

CAPACITANCE (TA = +25°C, F = 1.0MHz)

Symbol	Parameter(1)	Conditions	Typ.	Unit
CIN	Input Capacitance	VIN = 0V	10	pF
COUT	Output Capacitance	VOUT = 0V	12	pF

NOTE: 2562 tbl 07

1. This parameter is sampled at initial characterization and is not 100% tested.

TEST CIRCUIT

Test	Switch
tPLZ	Closed
tPZL	Closed
Open Drain	Closed
All Other Tests	Open

DEFINITIONS: 2562 tbl 10

CL = Load capacitance includes jig and probe capacitance.
RT = Termination should be equal to ZOUT of the pulse generator. (Typically 50Ω)
VIN = 0V to 3.0V
INPUT: tr = tf = 2.5ns (10% to 90%) unless otherwise specified

6

DC ELECTRICAL CHARACTERISTICS

Commercial: 0°C to +70°C, 5V ± 5%; Military: –55°C to +125°C, 5V ± 10%

Symbol	Parameter	Test Condition		Min.	Max	Unit
V_{IH}	High-Level Input Voltage	—		2.0	—	V
V_{IL}	Low-Level Input Voltage	—		—	0.8	V
I_{IH}	High Level Input Current	V_{CC} = Max.	V_I = V_{CC}	—	10	µA
I_{IL}	Low-Level Input Current	V_{CC} = Max.	V_I = GND	—	–10	µA
V_{OH}	High-Level Output Voltage	V_{CC} = Min., I_{OH} = -8mA(COM'L.), –6mA(MIL.)		2.4	—	V
V_{OL}	Low-Level Output Voltage	V_{CC} = Min., I_{OL} = 16mA(COM'L.), 12mA(MIL.)		—	0.4	V
V_{IK}	Input Clamp Voltage	I_I = –18mA		—	–1.2	V
I_{OS}	Short Circuit Output Current[2]	V_{CC} = Max., V_O = GND, V_I = V_{CC} or GND		–20	—	mA
I_{OZH}	High Impedance Output Current	V_{CC} = Max.	V_I = V_{CC}	—	20	µA
I_{OZL}	Low Impedance Output Current	V_{CC} = Max.	V_I = GND	—	–20	µA

NOTES:

2562 tbl 08

1. For conditions shown as Min. or Max., use appropriate value based on temperature range.
2. Not more than one output should be shorted at one time. Duration of the short circuit test should not exceed 100 milliseconds.

POWER SUPPLY CHARACTERISTICS

Symbol	Parameter	Test Conditions[1]		Min.	Typ.[2]	Max.	Unit
I_{CCQC}	Quiescent Power Supply Current	V_{CC} = Max., V_I = V_{LC} or V_{HC}		—	2	10	mA
I_{CCQT} [3]	Quiescent Power Supply Current Inputs HIGH	V_{CC} = Max., V_I = 3.4V		—	15	45	mA
I_{CCD1}[4]	Dynamic Power Supply Current	V_{CC} = Max., Outputs Disabled, $\overline{OE}$ = HIGH, f_{CP} = 10MHz, 50% Duty Cycle, $V_I \leq V_{HC}$, $V_I \geq V_{LC}$	COM'L.	—	10	30	mA
			MIL.	—	10	40	
I_{CCD1}[4]	Dynamic Power Supply Current	V_{CC} = Max., Outputs Disabled, $\overline{OE}$ = HIGH, f_{CP} = 40MHz, 50% Duty Cycle, $V_I \leq V_{HC}$, $V_I \geq V_{LC}$	COM'L.	—	10	60	mA
			MIL.	—	10	80	

NOTES:

2562 tbl 09

1. For conditions shown as Min. or Min., use appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical values are at V_{CC} = 5.0V, +25°C ambient and maximum loading, not production tested.
3. This parameter is not directly testable but is derived for use in the total power supply calculation.
4. $I_C = I_{QUIESCENT} + I_{INPUTS} + I_{DYNAMIC}$
   $I_C = I_{CCQC} + (I_{CCQT} \times D_H \times N_T) + I_{CCD}$
   I_{CCQC} = Quiescent Current
   I_{CCQT} = Power Supply Current for a TTL High Input (V_{IN} = 3.4V)
   D_H = Duty Cycle for each TTL Input High
   N_T = Number of TTL Inputs at D_H
   I_{CCD} = Dynamic Charge moved by an input transition pair (HLH or LHL)
   All currents are in milliamps and all frequencies are in megahertz.

AC ELECTRICAL CHARACTERISTICS

Commerical: T_A = 0°C to +70°C, V_{CC} = 5V ±5%; Military: T_A = –55°C to +125°C, V_{CC} = 5V ±10%

Parameter	Commercial				Military				Unit
	73200L12 73201L12		73200L15 73201L15		73200L15 73201L15		73200L20 73201L20		
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
CLK to Y_0–Y_{15} Propagation Delay	—	12	—	15	—	15	—	20	ns
SEL_0–SEL_2 to Y_0–Y_{15} Propagation Delay	—	12	—	15	—	15	—	20	ns
D_0–D_{15} to CLK Set-up Time	3	—	4	—	4	—	5	—	ns
D_0–D_{15} to CLK Hold Time	1	—	2	—	2	—	3	—	ns
I_0–I_3 to CLK Set-up Time	4	—	5	—	5	—	6	—	ns
I_0–I_3 to CLK Hold Time	2	—	2	—	2	—	3	—	ns
CEN to CLK Set-up Time	4	—	5	—	5	—	6	—	ns
CEN to CLK Hold Time	2	—	2	—	2	—	3	—	ns
$\overline{OE}$ Enable Time[1]	—	9	—	10	—	10	—	13	ns
$\overline{OE}$ Disable Time[1]	—	8	—	9	—	9	—	13	ns
CLK Pulse Width HIGH	5	—	5	—	5	—	6	—	ns
CLK Pulse Width LOW	5	—	5	—	5	—	6	—	ns
CLK Period	—	12	—	15	—	15	—	20	ns
Data In to Data Out Flowthrough[2]	—	12	—	15	—	-15	—	20	ns

NOTES: 2562 tbl 11

1. Output Enable and Disable times measured to 500mV change of output voltage level.
2. 73201 only.

AC TEST CONDITIONS

Input Pulse Levels	GND to 4.0V
Input Rise/Fall Times	4ns
Input Timing Reference Levels	1.5V
Output Reference Levels	1.5V
Output Load	See Figure 1

2562 tbl 12

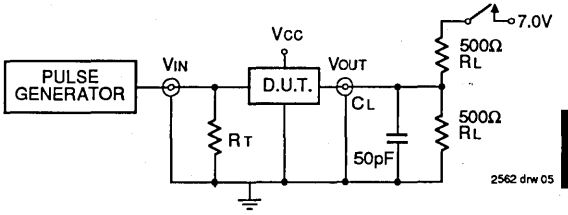


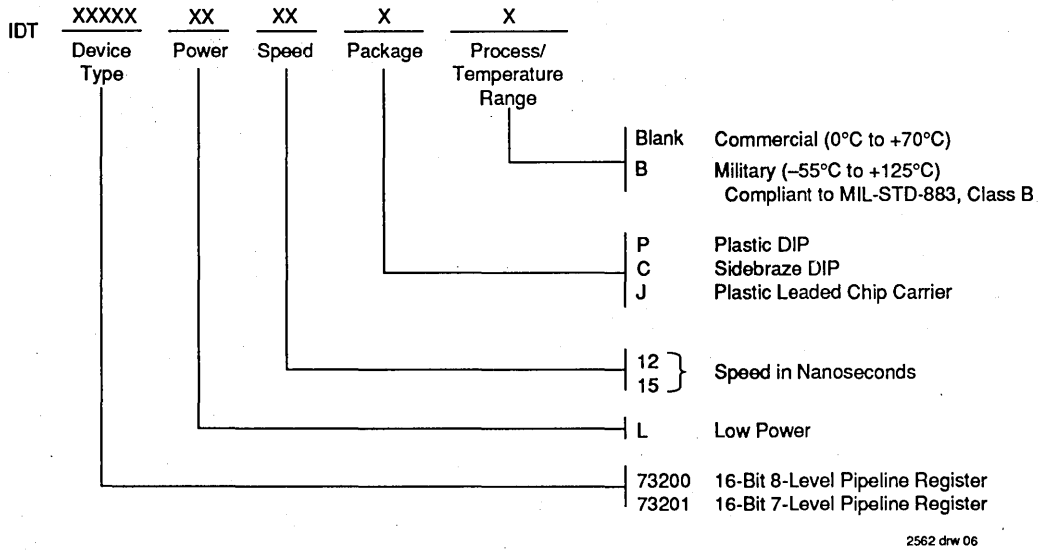
Figure 1. AC Output Test Circuit

CMOS TESTING CONSIDERATIONS

There are certain testing considerations which must be taken into account when testing high-speed CMOS devices in an automatic environment. These are:

1) Proper decoupling at the test head is necessary. Placement of the capacitor set and the value of capacitors used is critical in reducing the potential erroneous failures resulting from large Vcc current changes. Capacitor lead length must be short and as close to the DUT power pins as possible.
2) All input pins should be connected to a voltage potential during testing. If left floating, the device may begin to oscilliate causing improper device operation and possible latchup.
3) Definition of input levels is very important. Since many inputs may change coincidentally, significant noise at the device pins may cause the V_{IL} and V_{IH} levels not to be met until the noise has settled. To allow for this testing/board induced noise, IDT recommends using $V_{IL} \leq 0V$ and $V_{IH} \geq 3V$ for AC tests.
4) Device grounding is extremely important for proper device testing. The use of multi-layer performance boards with radial decoupling between power and ground planes is required. The ground plane must be sustained from the performance board to the DUT interface board. All unused interconnect pins must be properly connected to the ground pin. Heavy gauge stranded wire should be used for power wiring and twisted pairs are recommended to minimize inductance.

ORDERING INFORMATION



Integrated Device Technology, Inc.

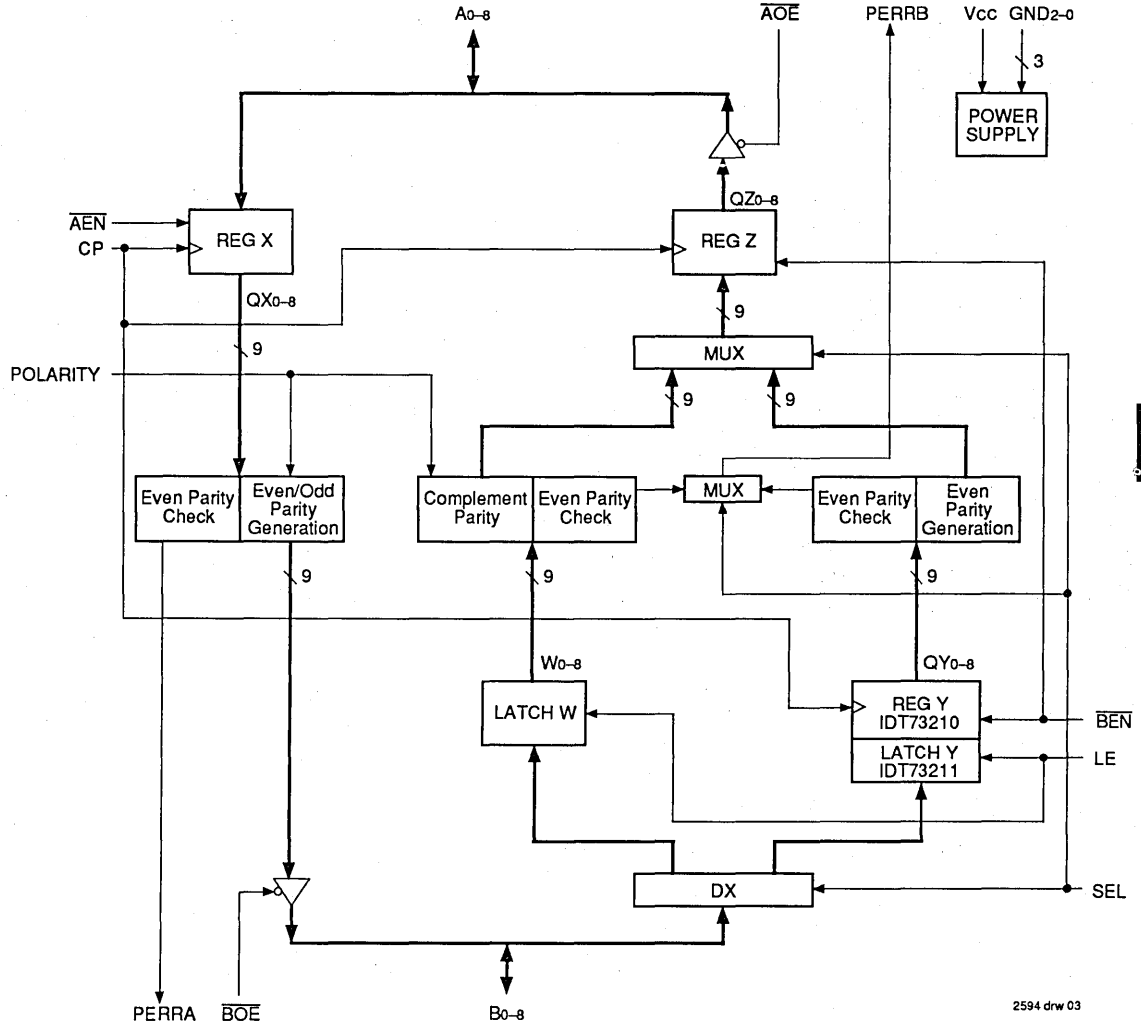
FAST CMOS OCTAL REGISTER TRANSCEIVER WITH PARITY

PRELIMINARY
IDT73210
IDT73211

FEATURES:

- Two bidirectional interfacing ports
- Single-level pipeline register for one port and one-level (73211) or two-level (73210) pipeline register for the other port
- 8-bit wide interface ports plus parity bit
- Even parity checking in both directions
- Even/odd parity generation from Port A to Port B
- Even parity generation from Port B to Port A
- Parity polarity control
- High output drive capability: 64/48mA (commercial/ military)
- Available in 32-pin, 300 mil plastic DIP and sidebraze DIP, surface mount 32-pin SOJ and LCC packages
- High-speed, low-power, CEMOS™ process technology
- Military product compliant to MIL-STD-883, Class B

FUNCTIONAL BLOCK DIAGRAM



CEMOS is a trademark of Integrated Device Technology Inc.

APPLICATIONS

- Cache memory bus interface
- Read and write buffers for RISC microprocessor system
- Registered transceiver with parity

FUNCTIONAL DESCRIPTION

The IDT73210/1 Octal Register Transceivers are high-speed, low-power data interface with data integrity checking capability.

They are designed for high-performance systems requiring bidirectional data transfer between two buses and maintaining error checking via parity.

In any RISC or CISC microprocessor system, the IDT73210/1 can be used to interface cache memory with main memory. Data integrity is ensured through parity checking. Control features allow dynamic reconfiguration of check/generate and odd/even parity options.

Detailed Functional Description

Port A to Port B Path (IDT73210 and IDT73211) is comprised of a register (X), an even/odd parity generator and an even parity checker. The input data is on the A_{0-8} lines. When $\overline{AEN}$ is low, A_{0-8} is latched into Register X on the low-to-high CP transition. Even parity of the latched data is checked. If PERRA goes high, a parity error has occurred. A new parity bit, B_8, is generated. The output data bus is B_{0-8} and is enabled when $\overline{BOE}$ is low.

Port B to Port A Path (IDT73210) is comprised of a latch (W), two registers (Y and Z), an even parity generator/checker and a parity bit latch complementor. The input data bus is on the B_{0-8} lines.

When SEL is high, the incoming data is latched into Latch W. When LE is high, Latch W is transparent; when LE is low, Latch W is closed. The parity bit, B_8, can be complemented by the POLARITY pin. If POLARITY is low, the parity sense remains the same. If POLARITY is high, the parity sense is complemented. Parity is not generated in this path. Even parity of latched data is checked. If PERRB goes high, a parity error has occurred. When $\overline{BEN}$ is low, W_{0-8} is latched into Register Z on the low-to-high CP transition. The previous contents are held in Register Z if $\overline{BEN}$ is high or if there is no low-to-high CP transition. The output data bus is A_{0-8} and is enabled when $\overline{AOE}$ is low. When SEL is high, there is only a one clock cycle latency.

When SEL is low, the incoming data is latched into Register Y on the low-to-high CP transition, when $\overline{BEN}$ is low. Even parity of the registered data is checked. If PERRB goes high, a parity error has occurred. Even parity (QY_8) is generated on the contents in Register Y. When $\overline{BEN}$ is low, the contents of register Y are transferred to Register Z on the low-to-high CP transition. When $\overline{BOE}$ is low, the content of Register Z is made available at output Port A. When SEL is low, there is a two clock cycle latency.

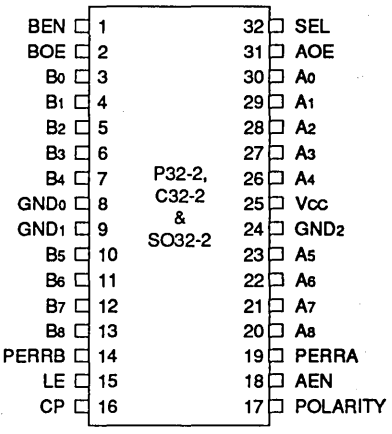
Port B to Port A Path (IDT73211) is comprised of a latch (W), two registers (Y and Z), an even parity generator/checker and a parity bit latch complementor. The input data bus is on the B_{0-8} lines.

When SEL is high, the incoming data is latched into Latch W. When LE is high, Latch W is transparent; when LE is low, Latch W is closed. The parity bit, B_8, can be complemented by the POLARITY pin. If POLARITY is low, the parity sense remains the same. If POLARITY is high, the parity sense is complemented. Parity is not generated in this path. Even parity of latched data is checked. If PERRB goes high, a parity error has occurred. When $\overline{BEN}$ is low, W_{0-8} is latched into Register Z on the low-to-high CP transition. The previous contents are held in Register Z if $\overline{BEN}$ is high or if there is no low-to-high CP transition. The output data bus is A_{0-8} and is enabled when $\overline{AOE}$ is low. When SEL is high, there is only a one clock cycle latency.

When SEL is low, the incoming data is latched into Latch Y when LE is high. Latch Y is closed when LE is low. Even parity of latched data is checked. If PERRB goes high, a parity error has occurred. Even parity (QY_8) is generated on the contents in Latch Y. When $\overline{BEN}$ is low, the contents of Latch Y are transferred to Register Z on the low-to-high CP transition. When $\overline{BOE}$ is low, the content of Register Z is made available at output Port A. When SEL is low, there is a one clock cycle latency.

The power pins are Vcc and GND_{0-2}. GND_0 is internal quiet ground, GND_1 is Port B ground and GND_2 is Port A ground.

PIN CONFIGURATIONS[1]



DIP/SOJ
TOP VIEW

NOTE:
1. GND_0 is internal quiet ground
   GND_1 is B Port ground
   GND_2 is A Port ground

PIN DESCRIPTIONS

Pin Name	I/O	Description
A_{0-8}	I/O	Data Port A.
$\overline{AEN}$	I	Clock enable (active low) for the register X.
$\overline{AOE}$	I	3-state output enable for Port A.
B_{0-8}	I/O	Data Port B.
$\overline{BEN}$	I	Clock enable (active low) for the registers Y and Z.
$\overline{BOE}$	I	3-state output enable for Port B.
LE	I	Latch enable input for Latch Y/Latch W of Port B. The Latch Y/Latch W is open when LE is high. Data is latched on the high-to-low transition of LE.
SEL	I	Input selection for Port B. SEL = 0 Register Y; SEL = 1 Latch Y
POLARITY	I	Polarity selection input. **Polarity** / **A to B Direction** / **B to A Direction** 0 / EVEN / Pass Parity 1 / ODD / Complement Parity
PERRA	O	Parity output error for Port A.
PERRB	O	Parity output error for Port B.
CP	I	Input clock.
VCC		+5 volts.
GND_{0-2}		Ground.

2594 tbl 01

6

OPERATING MODES SUMMARY

IDT73210/1 A TO B DIRECTION

Input	Reg. X	PERRA	Output (B8)	Output B0-8
A0-8	A0-8 → QX0-8 (CP = Lo to Hi) ($\overline{AEN}$ = 0)	Result of even parity check	Even/odd parity bit B8 = POLARITY XOR Even parity generate from QX0-7	QX0-8 → B0-8 ($\overline{BOE}$ = 0)

2594 tbl 02

IDT73210/1 B TO A DIRECTION WHEN SEL = 1

Input	Latch W	PERRB	Reg. Z (QZ8)	Reg. Z QZ0-8	Output (A8)	Output A0-8
B0-8	B0-8 → W0-8 (LE = 1)	Result of even parity check	Bit complemented by POLARITY (Even/odd parity translation)	W0-8 → QZ0-8 (CP = Lo to Hi) ($\overline{BEN}$ = 0)	A8 = POLARITY XOR W8	QZ0-8 → A0-8 ($\overline{AOE}$ = 0)

2594 tbl 03

IDT73210 B TO A DIRECTION WHEN SEL = 0

Input	Reg. Y	PERRB	Reg. Z (QZ8)	Reg. Z QZ0-8	Output (A8)	Output A0-8
B0-8	B0-8 → QY0-8 (CP = Lo to Hi) ($\overline{BEN}$ = 0)	Result of even parity check	Even parity generated bit	QY0-8 → QZ0-8 (CP = Lo to Hi) ($\overline{BEN}$ = 0)	A8 = Even parity generated from QY0-7	QZ0-8 → A0-8 ($\overline{BOE}$ = 0)

2594 tbl 04

IDT73211 B TO A DIRECTION WHEN SEL = 0

Input	Latch Y	PERRB	Reg. Z (QZ8)	Reg. Z QZ0-8	Output (A8)	Output A0-8
B0-8	B0-8 → QY0-8 (LE = 1)	Result of even parity check	Even parity generated bit	QY0-8 → QZ0-8 (CP = Lo to Hi) ($\overline{BEN}$ = 0)	A8 = Even parity generated from QY0-7	QZ0-8 → A0-8 ($\overline{BOE}$ = 0)

2594 tbl 05

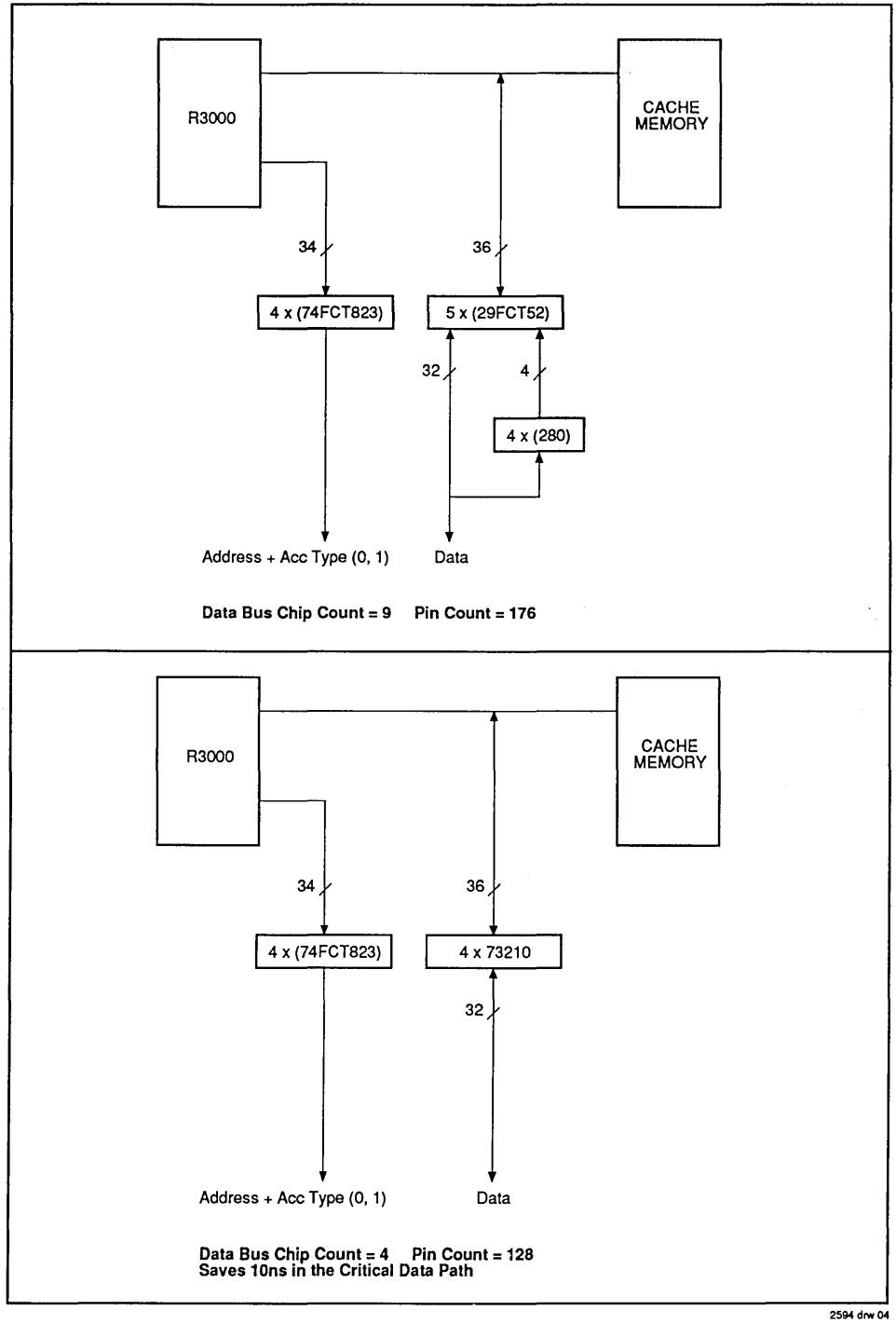


Figure 1. R3000 System with No Parity Support In Main Memory

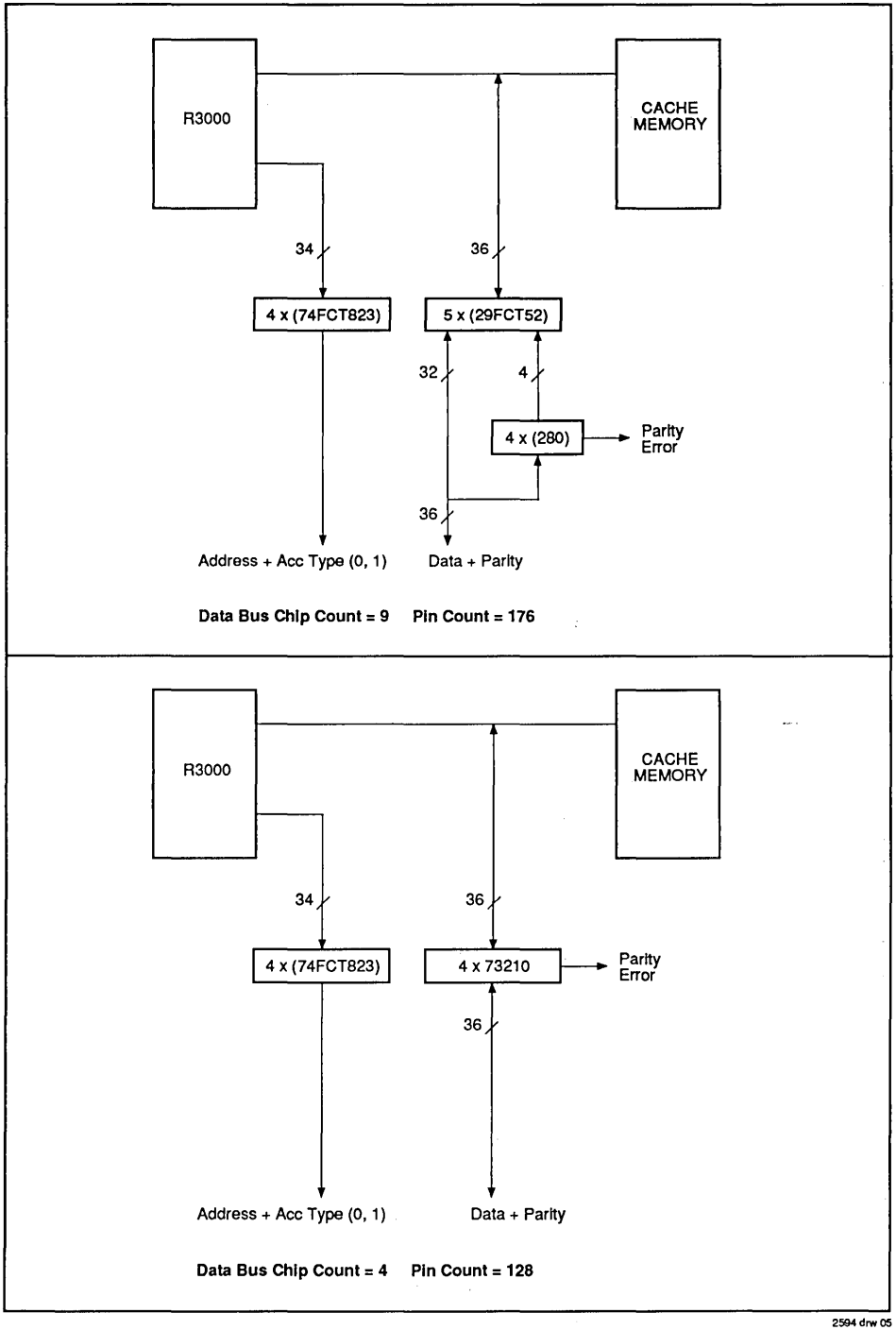


Figure 2. R3000 System with Parity Support in Main Memory

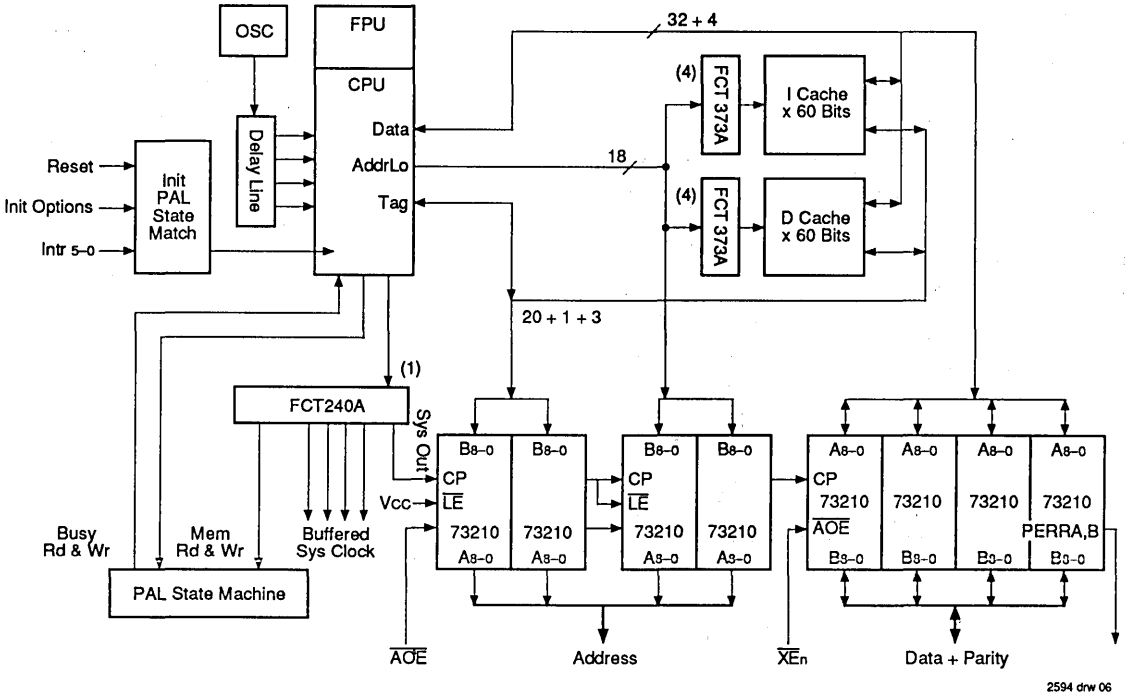


Figure 3. Read and Write Buffers Using Eight IDT73210/1

6

ABSOLUTE MAXIMUM RATINGS(1)

Symbol	Rating	Com'l.	Mil.	Unit
V_{TERM}	Terminal Voltage with Respect to Ground	–0.5 to V_{CC} + 0.5	–0.5 to V_{CC} + 0.5	V
V_{CC}	Power Supply Voltage	-0.5 to +7.0	-0.5 to +7.0	V
T_A	Operating Temperature	0 to +70	–55 to +125	°C
T_{BIAS}	Temperature Under Bias	–55 to +125	–65 to +135	°C
T_{STG}	Storage Temperature	–55 to +125	–65 to +150	°C
P_T	Power Dissipation	1.2	1.5	W
I_{OUT}	Total Output Current	200	250	mA

NOTE: 2594 tbl 06

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

CAPACITANCE (T_A = +25°C, f = 1.0MHz)

Symbol	Parameter(1)	Conditions	Typ.	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V	5	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V	7	pF
$C_{I/O}$	Input – Output Capacitance	V_{OUT} = 0V	7	pF

NOTE: 2594 tbl 07

1. This parameter is sampled and not 100% tested.

DC ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE

The following conditions apply unless otherwise specified:
Commercial: T_A = 0°C to +70°C, V_{CC} = 5.0V ± 5%; Military: T_A = –55°C to +125°C, V_{CC} = 5.0V ± 10%

Symbol	Parameter	Test Conditions(1)			Min.	Typ.(2)	Max.	Unit
V_{IH}	Input HIGH Level	Guaranteed Logic HIGH Level			2.0	—	—	V
V_{IL}	Input LOW Level	Guaranteed Logic LOW Level			—	—	0.8	V
I_{IH}	Input HIGH Current	V_{CC} = Max.		Except I/O	—	—	10	µA
		V_I = 2.7V		I/O pins	—	—	20	
I_{IL}	Input LOW Current	V_{CC} = Max.		Except I/O	—	—	-10	µA
		V_I = 0.5V		I/O pins	—	—	-20	
V_{IK}	Clamp Diode Voltage	V_{CC} = Min., I_N = –18mA			—	-0.7	-1.2	V
I_{OS}	Short Circuit Current	V_{CC} = Max.(3), V_O = GND		PERRA, PERRB	-30	—	-150	mA
				A_{0-8}, B_{0-8}	-20	—	-75	
V_{OH}	Output HIGH Voltage	V_{CC} = Min.		I_{OH} = –12mA MIL.	2.4	3.3	—	V
		V_{IN} = V_{IH} or V_{IL}		I_{OH} = –15mA COM'L.				
V_{OL}	Output LOW Voltage	V_{CC} = Min.	A_{0-8}	I_{OL} = 48mA MIL.	—	0.3	0.5	V
		V_{IN} = V_{IH} or V_{IL}	B_{0-8}	I_{OL} = 64mA COM'L.				
		V_{CC} = Min.	PERRA	I_{OL} = 20mA MIL.				
		V_{IN} = V_{IH} or V_{IL}	PERRB	I_{OL} = 24mA COM'L.				
V_H	Input Hysteresis for CP only	V_{CC} = 5V			—	200	—	mV

NOTES: 2594 tbl 09

1. For conditions shown as Min. or Max., use appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical values are at V_{CC} = 5.0V, +25°C ambient, not production tested.
3. Not more than one output should be shorted at one time. Duration of the short circuit test should not exceed 100 millisecond.

POWER SUPPLY CHARACTERISTICS

Commercial: T_A = 0°C to +70°C, V_{CC} = 5.0V ± 5%; Military: T_A = −55°C to +125°C, V_{CC} = 5.0V ± 10%
V_{LC} = 0.2V; V_{HC} = V_{CC} - 0.2V

Symbol	Parameter	Test Conditions[1]		Min.	Typ.[2]	Max.	Unit
I_{CCQC}	Quiescent Power Supply Current	V_{CC} = Max., V_{IN} = GND or V_{CC}		—	0.001	2.0	mA
I_{CCQT}	Quiescent Power Supply Current TTL Inputs HIGH	V_{CC} = Max. V_{IN} = 3.4[3]	COM'L.	—	3	10	mA
			MIL.	—	3	15	
I_{CCD1}	Dynamic Power Supply Current[4]	V_{CC} = Max. Outputs Disabled f_{CP} = 10MHz 50% Duty Cycle f_i = 5MHz	$V_{IN} \geq V_{HC}$ $V_{IN} \leq V_{LC}$	—	6.0	15	mA
I_{CCD2}	Dynamic Power Supply Current[4]	V_{CC} = Max. Outputs Disabled f_{CP} = 40MHz 50% Duty Cycle f_i = 20MHz	$V_{IN} \geq V_{HC}$ $V_{IN} \leq V_{LC}$	—	24	60	mA

2594 tbl 08

NOTES:
1. For conditions shown as Min. or Max., use appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical values are at V_{CC} = 5.0V, +25°C ambient and maximum loading, not production tested.
3. This parameter is not directly testable but is derived for use in the total power supply calculation.
4. $I_C = I_{QUIESCENT} + I_{INPUTS} + I_{DYNAMIC}$
   $I_C = I_{CCQC} + I_{CCQT} D_H N_T + I_{CCD}$
   I_{CCQC} = Quiescent Current
   I_{CCQT} = Power Supply Current for a TTL High Input (V_{IN} = 3.4V)
   D_H = Duty Cycle for TTL Inputs High
   N_T = Number of TTL Inputs at DH
   I_{CCD} = Dynamic Current caused by an Input Transition Pair (HLH or LHL)
   All currents are in milliamps and all frequencies are in megahertz.

6

SWITCHING CHARACTERISTICS OVER COMMERCIAL OPERATING RANGE

T_A = 0°C to +70°C; V_{CC} = 5V ± 5%
C_L = 50pF; R_L = 500Ω

Parameter	Description	Min.	Typ.[1]	Max.	Unit
t_{PLH} t_{PHL}	Propagation Delay Clock to A_{0-8} ($\overline{AOE}$ = Low) Clock to B_{0-8} ($\overline{BOE}$ = Low)	—	—	10.0	ns
t_{PHL}	Propagation Delay CP to PERRA, PERRB	—	—	8.5	ns
t_{PHL}	Propagation Delay POLARITY to B_{0-8}	—	—	7.0	ns
t_{PHL}	Propagation Delay B_{0-8} to PERRB LE = High	—	—	8.5	ns
t_S	Set-up Time A_{0-8}, B_{0-8}, POLARITY, SEL to CP	2.0	—	—	ns
t_H	Hold Time A_{0-8}, B_{0-8}, POLARITY, SEL to CP	1.5	—	—	ns
t_S	Set-up Time $\overline{AEN}$, $\overline{BEN}$ to CP Low-to-High	2.0	—	—	ns
t_H	Hold Time $\overline{AEN}$, $\overline{BEN}$ to CP Low-to-High	1.5	—	—	ns
t_S	Set-up Time B_{0-8} to LE	2.0	—	—	ns
t_H	Hold Time B_{0-8} to LE	1.5	—	—	ns
t_S	Set-up Time B_{0-8} to CP to Low-to-High; LE = High	3.0	—	—	ns
t_H	Hold Time B_{0-8} to CP to Low-to-High; LE = High	1.5	—	—	ns
t_{PZH} t_{PZL}	Output Enable Time $\overline{AOE}$ to A_{0-8}, $\overline{BOE}$ to B_{0-8}	—	—	7.0	ns
t_{PHZ} t_{PLZ}	Output Disable Time $\overline{AOE}$ to A_{0-8}, $\overline{BOE}$ to B_{0-8}	—	—	6.5	ns
t_{PWH}	Clock Pulse Width High	7.0	5.0	—	ns
t_{PWL}	Clock Pulse Width Low	7.0	5.0	—	ns

NOTE:
1. Typical values are at Vcc = 5.0V and +25°C ambient, not production tested.

2594 tbl 10

SWITCHING CHARACTERISTICS OVER MILITARY OPERATING RANGE

T_A = -55°C to +125°C; V_{CC} = 5V ± 10%
C_L = 50pF; R_L = 500Ω

Parameter	Description	Min.	Typ.(1)	Max.	Unit
t_{PLH} t_{PHL}	Propagation Delay Clock to A_{0-8} ($\overline{AOE}$ = Low) Clock to B_{0-8} ($\overline{BOE}$ = Low)	—	—	12.2	ns
t_{PHL}	Propagation Delay CP to PERRA, PERRB	—	—	10.6	ns
t_{PHL}	Propagation Delay POLARITY to B_{0-8}	—	—	7.0	ns
t_{PHL}	Propagation Delay B_{0-8} to PERRB LE = High	—	—	10.6	ns
t_S	Set-up Time A_{0-8}, B_{0-8}, POLARITY, SEL to CP	2.0	—	—	ns
t_H	Hold Time A_{0-8}, B_{0-8}, POLARITY, SEL to CP	1.5	—	—	ns
t_S	Set-up Time $\overline{AEN}$, $\overline{BEN}$ to CP Low-to-High	2.0	—	—	ns
t_H	Hold Time $\overline{AEN}$, $\overline{BEN}$ to CP Low-to-High	1.5	—	—	ns
t_S	Set-up Time B_{0-8} to LE	2.0	—	—	ns
t_H	Hold Time B_{0-8} to LE	1.5	—	—	ns
t_S	Set-up Time B_{0-8} to CP to Low-to-High; LE = High	3.0	—	—	ns
t_H	Hold Time B_{0-8} to CP to Low-to-High; LE = High	1.5	—	—	ns
t_{PZH} t_{PZL}	Output Enable Time $\overline{AOE}$ to A_{0-8}, $\overline{BOE}$ to B_{0-8}	—	—	7.0	ns
t_{PHZ} t_{PLZ}	Output Disable Time $\overline{AOE}$ to A_{0-8}, $\overline{BOE}$ to B_{0-8}	—	—	6.5	ns
t_{PWH}	Clock Pulse Width High	8	6	—	ns
t_{PWL}	Clock Pulse Width Low	8	6	—	ns

NOTE:
1. Typical values are at Vcc = 5.0V and +25°C ambient.

2594 tbl 11

6

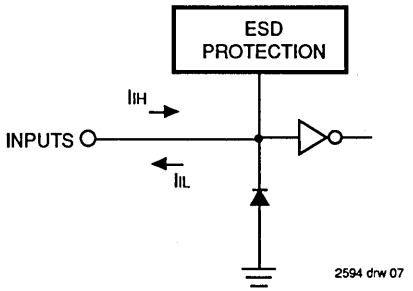


Figure 4. Input Interface Circuit

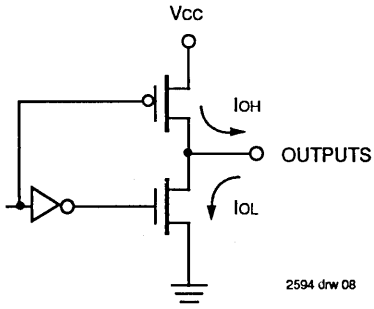
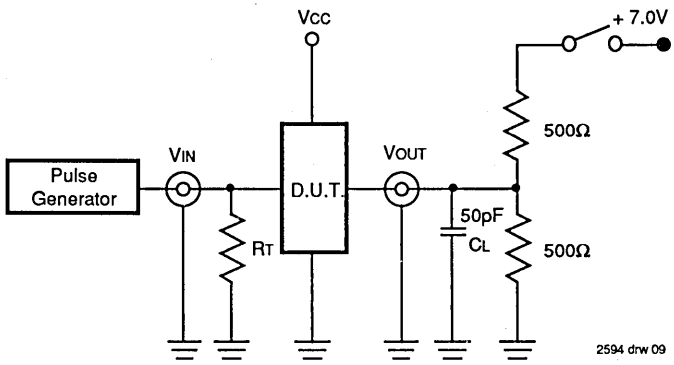


Figure 5. Output Interface Circuit



DEFINITIONS:
C_L = Load capacitance: includes jig and probe capacitance
R_L = Termination resistance: should be equal to Z_{OUT} of the Pulse Generator

Figure 6. AC Test Load Circuit

AC TEST CONDITIONS

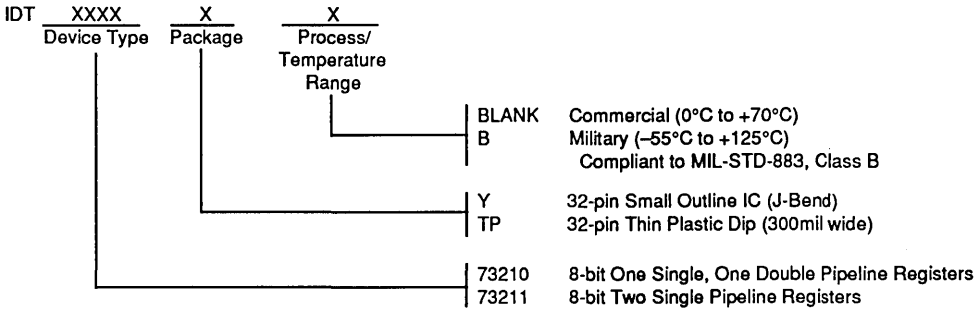
Input Pulse Levels	GND to 3.0V
Input Rise/Fall Times	1V/ns
Input Timing Reference Levels	1.5V
Output Reference Levels	1.5V
Output Load	See Figure 6

2594 tbl 12

Test	Switch
Open Drain Disable Low Enable Low	Closed
All other Tests	Open

2594 tbl 13

ORDERING INFORMATION



2594 drw 10

6

RISC MODULE PRODUCTS

Maximizing the performance of R3000 systems means designing with very high-speed components and finely tuning PC board layouts to work at very high clock rates. IDT offers a variety of pre-built, pre-tested RISC subsystems that can be used to eliminate this part of the design task.

Roughly three by six inches in size, the modules are built on 8-10 layer PC boards with components surface mounted on both sides. Most modules include at least the CPU, optional FPA, cache RAMs and Read/Write Buffers. The high-speed clock is also on the module, along with some reset, interrupt and control logic. The net effect is to put all of the very high-speed logic onto a tightly integrated, independent subsystem that can be purchased like a component. The modules are 100% burned-in and tested at the rated speed.

The modules are designed around several different architectures. Within a given architecture, there is a range of speeds and cache sizes to choose from. All the modules with the same architecture are plug compatible, so price/performance options are easy to offer in the end system by simply selecting the appropriate RISC SubSystem module.

Prototyping development systems are available for each module architecture to serve as a starting point for additional hardware and software development.

TABLE OF CONTENTS

Integrated Device Technology, Inc.

R3000 CPU MODULES FOR GENERAL APPLICATIONS

IDT7RS101

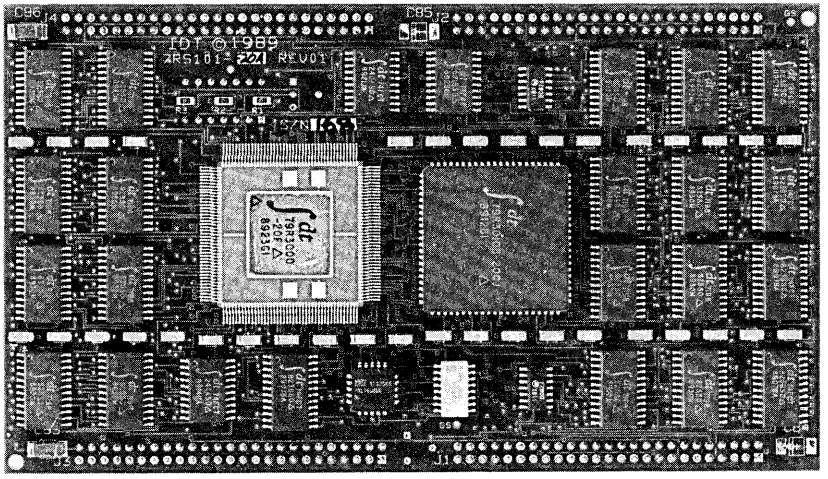
FEATURES:

- R3000 CPU on 3.7" x 6.5" plug-in module
- 64K each Instruction and Data Caches
- On–board clock generation
- Four-word read buffer for block refill. Single word write buffer
- On–board parity generation
- Five user interrupts into on-board register
- Available with or without Floating Point Accelerator
- Cache supports full 32-bit address space
- 100% burn-in and functional test at rated speed

POWERFUL GENERAL PURPOSE R3000 MODULE:

The IDT7RS101 is a complete reduced instruction set computer (RISC) CPU, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the R3000 CPU, 64 Kbytes each of data and instruction cache memory, a single word write buffer and a four-word read buffer to support block refill. Clock generation, reset, control and interrupt functions are included on the module to simplify the remainder of the system design. Parity bits on incoming data words may be generated automatically on the module, transparent to the rest of the system. Alternatively, parity may be handled in the user system, with on-board circuits only performing optional parity check functions. Five user interrupts are provided with an on-board clocked register to ensure synchronized activity with the R3000 timings.

The module is constructed using surface mount devices on a 3.72 by 6.5" epoxy laminate board, and is connected to the user's system via four 50-pin Insulation Displacement Connectors.



7RS101 Module. Actual Size 3.72" x 6.50 "

7

ARCHITECTURAL HIGHLIGHTS

Four-word Read Buffer

The 7RS101 includes a single write buffer and a four-word deep read buffer. Cache read miss operations (memory data not currently stored in the cache) may be handled either with a four-word block refill or with a single-word fetch and cache update. All control signals are available to implement either option. Address mapping can be used to force block refills on some addresses (for example, instructions) and single-word updates on other addresses (for example, data).

Clock Generation

The clocks for both the R3000 and the R3010 are automatically generated on the module using a very accurate and stable delay line driven by a single user-supplied input clock signal. There are three buffered clock output signals for use with external control logic and system timing, each of which is an identical inverted version of the R3000 output, SYSOUT#.

Parity Generation

The R3000 Processor requires incoming data words to consist of 32 bits of data and 4 bits of parity. The 7RS101 module can be set to either of two modes for parity handling. It can check parity and report errors on incoming data that consists of 32 data bits and 4 parity bits. In the other mode, it can generate parity on 32 bits of incoming data and supply the full 36 bits to the CPU.

Initialization Options

The R3000 requires mode selection to be made during the RESET initialization sequence. The 7RS101 module provides three pins that can simply be tied High or Low by the user to select some of the R3000's options: Instruction Streaming on or off, Partial Word Store on or off, and BigEndian or LittleEndian byte order.

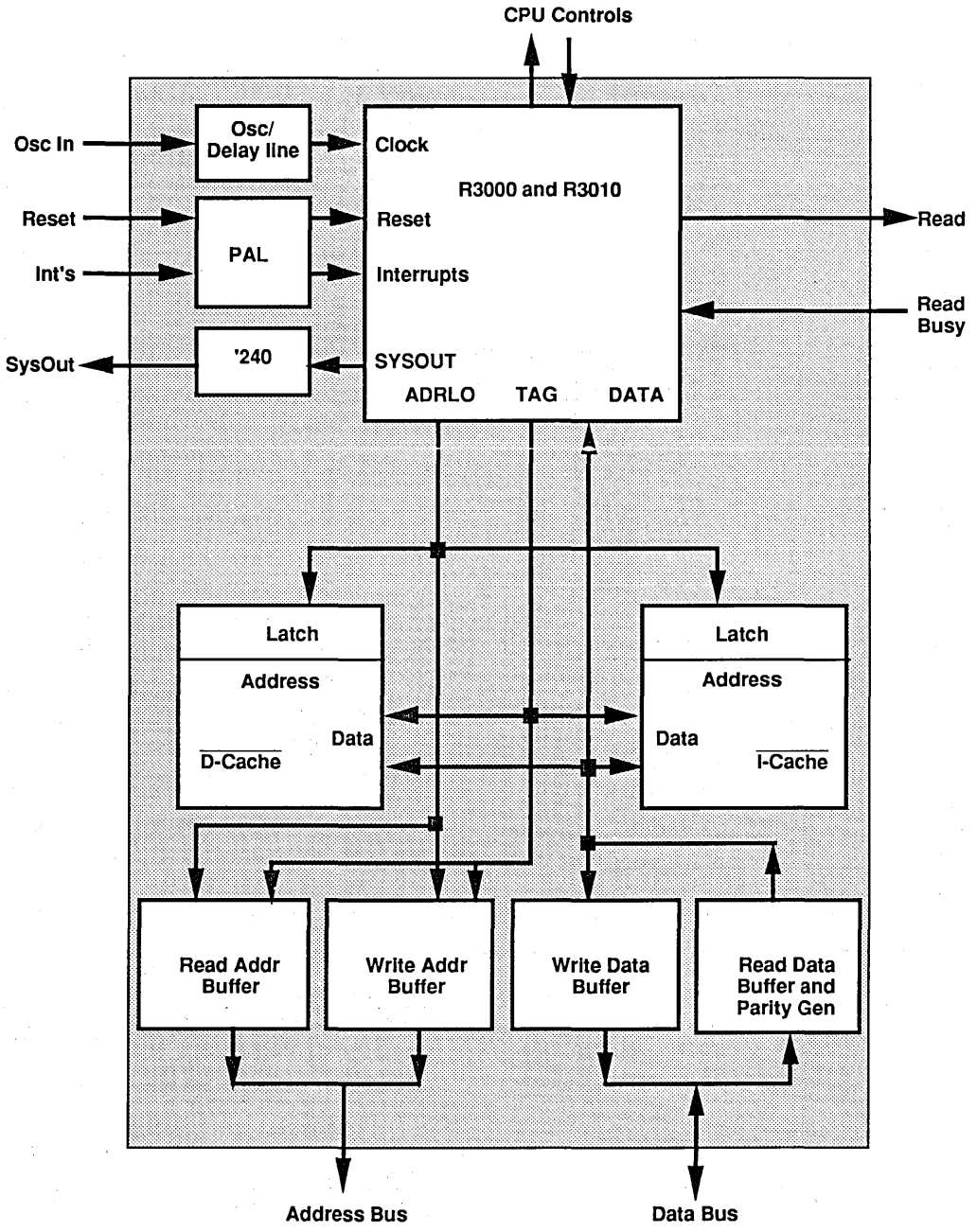
User Interrupts

Six user interrupt inputs are provided. Each of these is a negative-true signal, terminated with a 4.7K ohm pull-up load resistor on the module, so pins may be left unconnected if they are not used. The interrupt signals are clocked into an Interrupt Input Register on the module by the CPU clock SYSOUT. This ensures that the interrupt inputs to the R3000 are synchronized to its clock. One of the interrupts is a reserved for use by the R3010 FPA; the other 5 are available for the user.

Buffered Outputs

The address and data lines coming out of the module are buffered, and can support substantial bus drive. All control signals except those coming directly from the R3000 or R3010 are also buffered.

FUNCTIONAL BLOCK DIAGRAM



7

SIGNALS PROVIDED ON MODULE PINS

Signal	Type	Functional Description
MD00-MD31	I/O	Memory data lines to/from main memory system.
PAR0-PAR3	I/O	Parity bits for data lines. Unconnected if on-board parity generation selected.
MA00-MA31	OUT	Memory address lines to main memory system. These are registered outputs.
CA2-CA3	OUT	Block refill counter outputs. These lines are normally used instead of MA02-MA03, since they are the outputs of the counter used to implement the 4-word block refill function.
RACT(0:1)	OUT	Positive-true outputs indicating the states of the R3000 outputs,ACCTYP(0:1), which identify the size of data transactions for read/write cycles. These are registered outputs, like MA00-MA31.
BACT2	OUT	Buffered ACCTYP(2) output from R3000, used to distinguish between cache and non-cache memory operations.
AOE#	IN	Negative-true input to enable the 3-state register output pins, MA00-MA31 and RACT(0:1).
UINT0-UINT5	IN	User interrupt inputs. Each has a 4.7K ohm pullup load resistor. UINT1 is reserved for R3010 FPA usage.
OSCIN	IN	Oscillator input clock signal. (2x clock rate).
MRES#	IN	Negative-true master reset input.
RESSW1-3	IN	Mode selection inputs used to determine R3000 setup options during reset initialization sequence. Each has a 4.7K ohm pullup load resistor. Jumpers or switches to ground select the desired options.
SYSOUT1-3	OUT	Buffered clock outputs for synchronizing external events. Each of these is an identical clock signal, representing the inverted form of the R3000 output, SYSOUT#.
MEMRD#	OUT	Direct negative-true output from R3000, used to indicate that a memory read cycle is in progress.
MEMWR#	OUT	Direct negative-true output from R3000, used to indicate that a memory write cycle is in progress.
RBSY	IN	Positive-true input used to request a memory read stall initiation and termination. This signal is normally held in its asserted state and de-asserted at the completion of CPU stalls.
WBSY#	IN	Negative-true input used to request a busy indication for subsequent memory write operations.
BLKR#	IN	Negative-true input used to request a block read sequence for read operations from main memory.
AEN#	IN	Negative-true input used to enable the clock for loading the address register.
CEN	IN	Positive-true input used to enable the increment of the block refill address counter for pins CA2-CA3.
PHOLD	IN	Positive-true input used to inhibit clocking of the read buffer. This signal is normally the complement of the CEN input.
WOE#	IN	Negative-true input used to enable the data output drivers for main memory write cycles.
WCTL#	IN	Negative-true input used to enable the clock to load data into the write data register for main memory write cycles.
CPC0	IN	Direct input to R3000 Processor, used to indicate the size of the data (block, word, byte, or other) for memory read cycles.
CPC1	OUT	Connection between the R3000 Processor and the R3010 FPA. Indicates the status of the conditional branch. This pin is provided for diagnostic purposes, only.
EXC#	OUT	Direct output from R3000, indicating the EXC# signal between the R3000 and the R3010.
RUN#	OUT	Negative-true output from the R3000, indicating that the R3000 is in its RUN state (not stalled).
BERR#	IN	Negative-true input to R3000 (with 4.7K ohm pullup), indicating a bus error in main memory.
FPA#	OUT	Negative-true output from R3010, indicating the presence of R3010 FPA on the module.
POE#	IN	Negative-true input with 4.7K ohm pullup resistor, used to enable the on-board parity generation logic. It is left unconnected if parity is to be handled by user system.
PERR#	OUT	Negative-true output which indicates a parity error on incoming data when on-board parity generation is not selected.
TAGV	OUT	Tag validity indicator, connected between R3000 and cache memory. Provided for diagnostic purposes, only.

RELATED PRODUCTS

Prototyping System

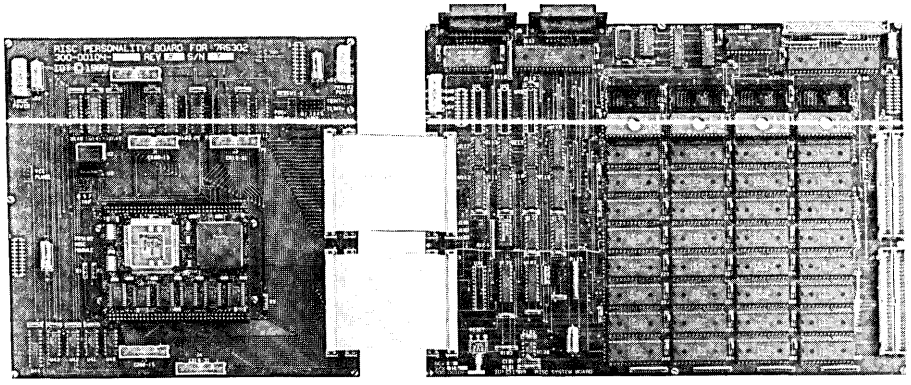
The 7RS101 module can be placed into immediate service using our flexible 7RS301 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's Software Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's PC-based cross assembler and compiler products. Assembled code can be downloaded into the Prototyping System for execution and debug.



ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed	Other
7RS101F66A16A	R3000A	R3010A	64K	64K	16 MHz	
7RS101F66A20A	R3000A	R3010A	64K	64K	20 MHz	
7RS101F66A25A	R3000A	R3010A	64K	64K	25 MHz	
7RS101F66A30A	R3000A	R3010A	64K	64K	30 MHz	
7RS101N66A16A	R3000A	None	64K	64K	16 MHz	
7RS101N66A20A	R3000A	None	64K	64K	20 MHz	
7RS101N66A25A	R3000A	None	64K	64K	25 MHz	
7RS101N66A30A	R3000A	None	64K	64K	30 MHz	

ADDITIONAL INFORMATION

For detailed technical specifications on this module refer to the 7RS101 Product Specification and User's Manual.

CUSTOM OPTIONS

Some features of the 7RS101 can be modified on special order. Contact your IDT sales office for information.

Integrated Device Technology, Inc.

R3000 CPU MODULES FOR COMPACT SYSTEMS

IDT7RS102

FEATURES:

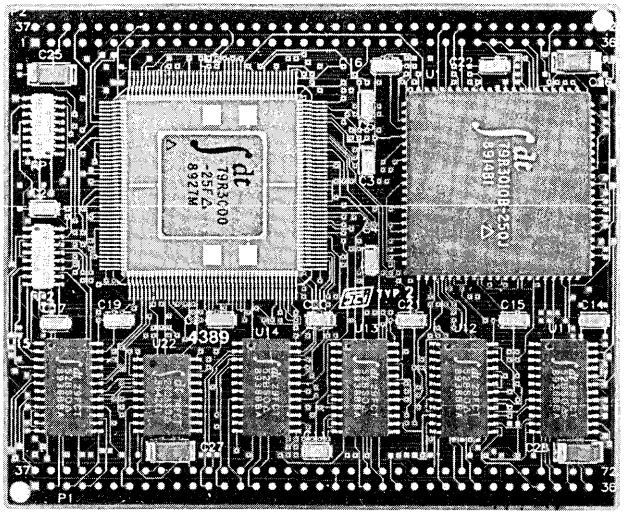
- **Cache Size: 16K Instruction, 16K Data**
- **Extremely small size: 3.2" x 3.9"**
- **Processor Speeds up to 25 MHz**
- **Includes R3010 Floating Point Accelerator**
- **Single word Read and Write Buffers**
- **100% burn-in and functional test at rated speed**

R3000 MODULE FOR GENERAL USE IN SMALL SYSTEMS:

The IDT7RS102 is a complete reduced instruction set computer (RISC) CPU, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the R3000 CPU, the R3010 Floating Point Accelerator,16 Kbytes each of data and instruction cache memory, a single word read buffer and a single word write buffer.

Cache misses are handled with single word read requests to memory, providing a simple interface to any type of main memory system.

The module is constructed using surface mount devices on a 3.2" by 3.9" epoxy laminate board, and is connected to the user's system via 144 pins located in two pin row regions on the board.



7RS102 Module. Actual Size 3.2" x 3.9 "

ARCHITECTURAL HIGHLIGHTS

Small and Simple

The 7RS102 module is designed to be as small as possible and to provide a simple interface to the user's system. The 16K caches are the smallest useful for most systems.

R3010 Floating Point Accelerator

The R3010 Floating Point Accelerator (FPA) is included as an integral part of the module. It operates in conjunction with the R3000 RISC Processor and greatly improves the system performance by expanding the instruction set to include very fast floating point capabilities. All timing and control connections are on the module and are completely transparent to the user.

Clock Generation

The clock inputs to the 7RS102 are the direct connections to the clocks for both the R3000 and the R3010. These clocks must be generated in the user system and applied to the module.

Cache Memory

Cache memory is provided on the module for a capacity of 16K bytes for each of the two R3000 cache memory systems (Instruction Cache and Data Cache). Memory operations which require main memory data transfers are conveniently handled by means of a variety of on-board control signals.

Cache read miss operations are handled as single-word fetch and cache update. Non-cache read operations (such as I/O reads) are indicated by means of control signals and are easily accomodated by the user.

Parity Generation

The R3000 Processor requires incoming data words to consist of 32 bits of data and a 4-bit parity code. Each of the 4 parity bits applies to a particular byte in the word. The required parity is even. The user system is required to generate parity for incoming data to the module and may optionally check parity for data being passed to main memory.

Address and Data Buffers

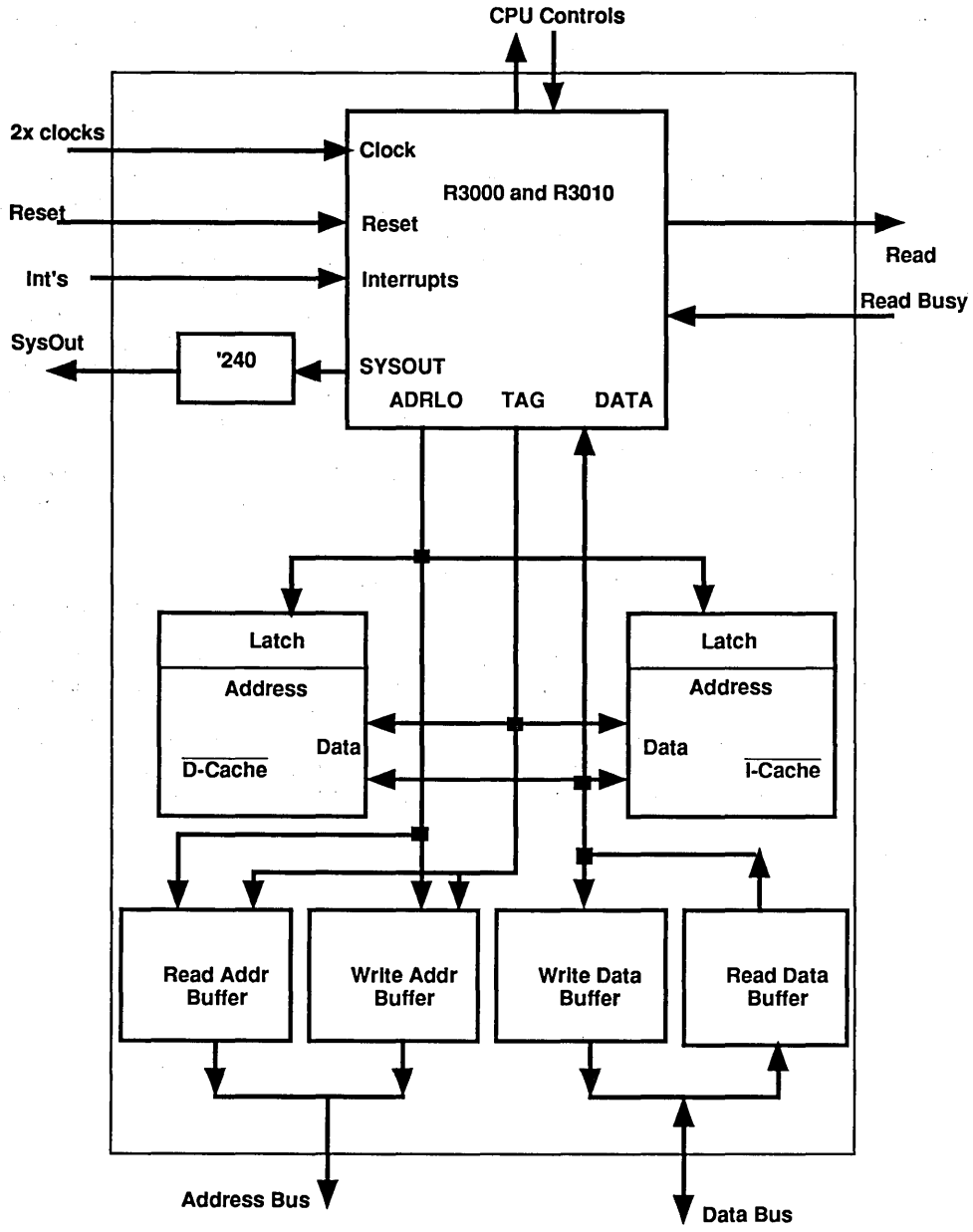
The address and data lines coming out of the module are buffered and can support substantial drive requirements. The address pins are direct outputs from registers and include the signals MACT0-MACT2. The three-state output drivers may be disabled by de-asserting the output enable control line, AOE. This is not normally done, but is provided as a feature for systems which may require it.

The data pins are driven by Bi-directional Registers. Enable/disable control of the three-state output drivers is accomplished with the signal, DOE. Memory write cycles utilize a single-word write buffer on the module which permits the R3000 Processor to continue running while data is being written into main memory. USER INTERRUPT INPUTS

R3000 User Interrupts

Six user interrupt inputs are provided, INT0-INT5. Each of these is a negative-true signal, terminated with a 10K ohm pullup load resistor on the module. In this way, the pin may be left unconnected if it is not to be used. The interrupt signals are connected directly to the interrupt pins of the R3000 Processor. INT1 is a reserved pin on this version of the module and is required for use by the R3010 FPA. As a result, it may not be used and must be left unconnected.

FUNCTIONAL BLOCK DIAGRAM



SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
MD00-MD31	I/O	Memory data lines to/from main memory system.
MPAR0-MPAR3	I/O	Parity bits for data lines. Parity must be supplied to the 7RS102 module, in accordance with R3000 requirements.
MA00-MA31	OUT	Memory address lines to the main memory system. These are registered outputs.
MACT0-MACT2	OUT	Positive-true outputs indicating the states of the R3000 outputs, ACCTYP(0:2), which identify the nature of the data transactions for read/write cycles. These are registered outputs, like MA00-MA31.
MTAGV	OUT	Registered TAGV output from R3000.
ACCTYP2	OUT	Unbuffered ACCTYP(2) output from R3000, used to distinguish between cache and non-cache memory operations.
AOE#	IN	Negative-true input to enable the three-state register output pins, MA00-MA31, MACT0-MACT2, and MTAGV.
INT0-INT5	IN	Interrupt inputs. Each has a 10K ohm pullup loadresistor. INT1 is reserved for R3010 FPA usage.
CLK2XPHI	IN	Clock input for R3000 and R3010. Timings must conform to R3000 specifications.
CLK2XRD	IN	Clock input for R3000 and R3010. Timings must conform to R3000 specifications.
CLK2XSYS	IN	Clock input for R3000 and R3010. Timings must conform to R3000 specifications.
CLK2XSMP	IN	Clock input for R3000 and R3010. Timings must conform to R3000 specifications.
MRES#	IN	Negative-true master reset input. Connects directly to R3000 RES# input pin.
SYSOUT1-3	IN	Buffered clock outputs for synchronizing external events. Each of these is an identical clock signal, representing the inverted form of the R3000 output, SYSOUT#.
MEMRD#	OUT	Direct negative-true output from R3000, used to indicate that a memory read cycle is in progress.
MEMWR#	OUT	Direct negative-true output from R3000, used to indicate that a memory write cycle is in progress.
RBSY	IN	Positive-true input used to request a memory read stall initiation and termination. This signal is normally held in its asserted state and deasserted at the compl;etion of stalls.
WBSY#	IN	Negative-true input used to request a busy indication for subsequent memory write operations.
ACE#	IN	Negative-true input used toenable the clock for loading the address register.
RDEN#	IN	Negative-true input used toenable the clock for loading the data register for memory read cycles..
DOE#	IN	Negative-true input used to enable the three-state data outputs, MD00-MD31 and MPAR0-MPAR3.
WCTL#	IN	Negative-true input used to enable the clock to load data into the write data register for main memory write cycles.
CPC0	IN	Direct input to the R3000, used to indicate tha size of the data (block, word, byte, or other) for memory read cycles.
CPC1	OUT	Connection between the R3000 Processor and the R3010 FPA, indicating the status of the conditional branch. This pin is provided for diagnostic purposes, only.
CPC2, CPC3	I/O	Direct connections to R3000 pins.
EXC#	OUT	Direct output from R3000, indicating the EXC# signal between the R3000 and the R3010.
RUN#	OUT	Negative-true output from the R3000, indicating that the R3000 is in its RUN state (not stalled).
BERR#	IN	Negative-true input to R3000, used to indicate a bus error in main memory.
FPA#	OUT	Negative-true output fro R3010, indicating the presence of R3010 FPA on the module.

RELATED PRODUCTS

Prototyping System

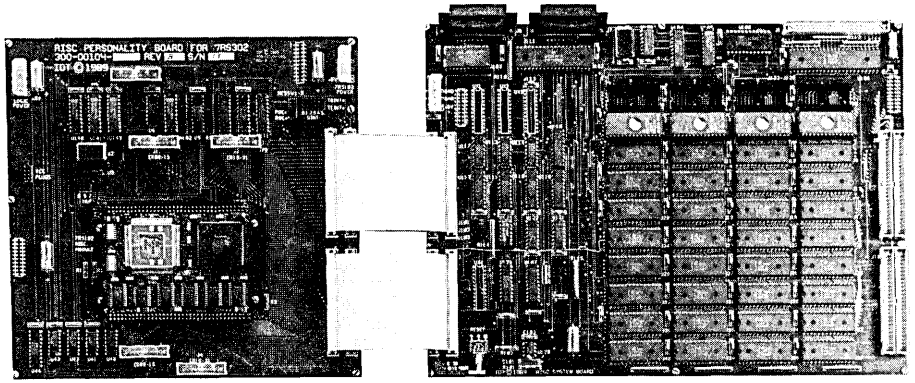
The 7RS102 module can be placed into immediate service using our flexible 7RS302 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's IDT/sim System Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's PC-based cross assembler and compiler products. Assembled code can be downloaded into the Prototyping System for execution and debug.



Module Prototyping Platform.
The card on the left is the personality card with a module; the card on the right is the general purpose CPU.

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed	Other
7RS102-16A	R3000	R3010	16K	16K	16 MHz	
7RS102-20A	R3000	R3010	16K	16K	20 MHz	
7RS102-25A	R3000	R3010	16K	16K	25 MHz	
7RS102F16A	R3000	R3010	16K	16K	16 MHz	
7RS102F20A	R3000	R3010	16K	16K	20 MHz	
7RS102F25A	R3000	R3010	16K	16K	25 MHz	

NOTE:
1. 7RS110 module recommended for new designs.

ADDITIONAL INFORMATION

For more details on the 7RS102 module, refer to the 7RS102 Technical Specification and User's Manual.

Integrated Device Technology, Inc.

R3000 CPU MODULES FOR COMPACT SYSTEMS

IDT7RS103

DISTINCTIVE FEATURES:

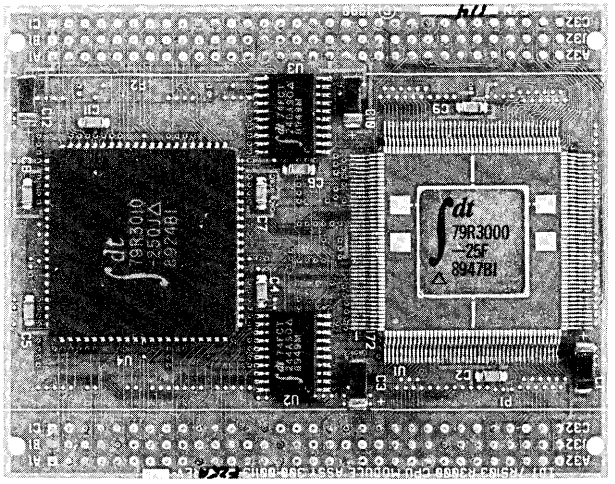
- Three Cache Size Versions:
  16K Instruction, 16K Data (7RS103-44)
  32K Instruction, 32K Data (7RS103-55)
  64K Instruction, 64K Data (7RS103-66)
- Extremely small size: 2.9" x 3.7"
- Processor speeds up to 25 MHz
- Optional R3010 Floating Point Accelerator
- On-board delay line to create R3000 clocks
- 100% burn-in and functional test at rated speed

R3000 MODULE FOR COMPACT HIGH PERFORMANCE SYSTEMS:

The IDT7RS103 is a family of interchangeable RISC CPU SubSystem modules, based on the MIPS R3000 RISC processor, and supplied on small fully-tested high-density plug-in modules. The module includes the R3000 CPU, optionally the R3010 Floating Point Accelerator, and data and instruction cache memory. Versions are available with 16K each I and D cache (7RS103-44), 32K each I and D cache (7RS103-55) and 64K each I and D cache (7RS103-66). The three versions differ only in the length of the board. All plug into the same socket. The delay line to generate the three R3000 2x clock signals is included on the module, so the module can be driven from a single 2x clock.

Externally, the user system supplies the R3000 control signals and the read and write buffers.

The module is constructed using surface mount devices on both sides of a 2.9" epoxy laminate board, and is connected to the user's system via 192 pins located in two pin row regions on the board.



7RS103-44 Module. Actual Size 2.9" x 3.7"

7

ARCHITECTURAL HIGHLIGHTS:

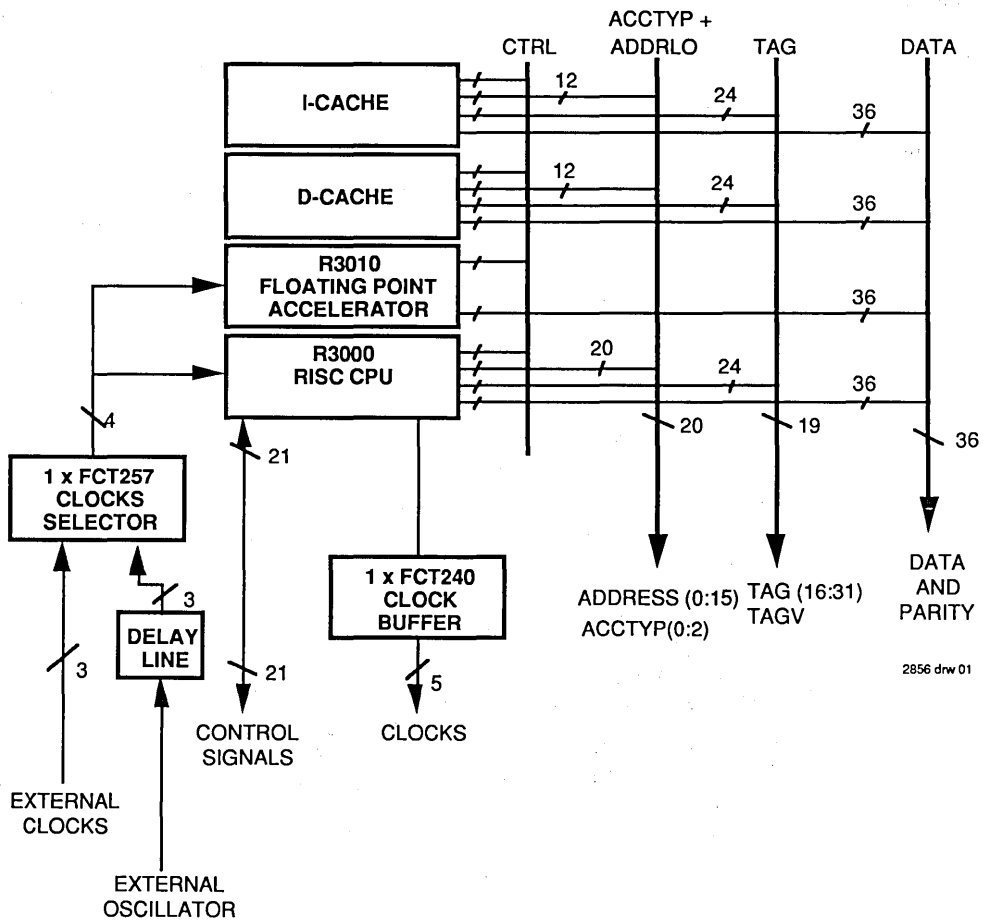
The Minimal Module

The 7RS103 is designed to provide an R3000 RISC SubSystem in as small a space as possible. It includes only the CPU (and FPA), cache memories, and a delay line to generate the 2x clocks to the R3000. The read and write buffers and control logic are handled off the module by the user's system. This makes the module ideal for use with ASICs or other unique implementations of main memory interface.

The R3000 timing and control signals are brought directly off the module. The R3000 data sheet should be consulted for all the timing specifications. One of the interrupt inputs is required by the R3010 on versions that include the FPA device.

The three versions of the 7RS103 differ only in the cache memory sizes. They are completely interchangeable.

FUNCTIONAL BLOCK DIAGRAM



Integrated Device Technology, Inc.

R3001 RISC ENGINE FOR EMBEDDED CONTROLLERS

ADVANCE INFORMATION
IDT7RS104

FEATURES:

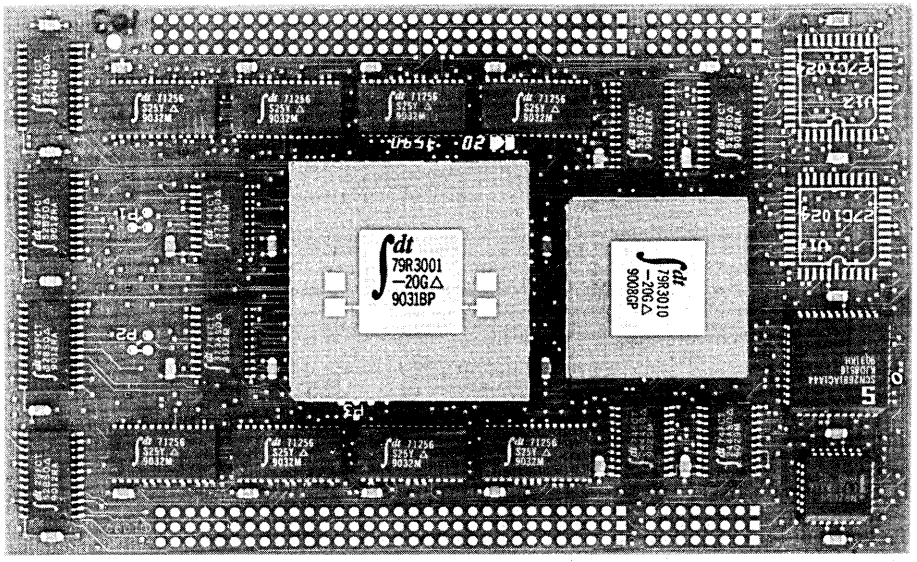
- 128K each of Instruction and Data RAM in Synchronous Memory
- Simple Interface to peripherals
- On-Board Dual UART
- On-Board DMA Control
- Includes IDT/sim Monitor in EPROM
- On-Board 8254 Counter/Timer
- On-board oscillator, delay line, and reset circuitry
- 100% burn-in and functional test at rated speed

R3001 MODULE WITH SYNCHRONOUS MEMORY

The IDT7RS104 is a complete reduced instruction set computer (RISC) CPU, based on the IDT 79R3001, an IDT derivative of the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the CPU, optionally the R3010 Floating Point Accelerator, and 256K of synchronous memory divided into 128K each for Instruction and data space. Clock generation, reset, control and interrupt functions are included on the module to simplify the remainder of the system design.

The 104 module takes advantage of the R3001's ability to address large amounts of synchronous memory, permitting the entire application program to reside in high-speed memory space.

The module is constructed using surface mount devices on a 3.7" by 6.1" epoxy laminate board, and is connected to the user's system via pins located in two pin row regions on the board.



7RS104 Module. Actual Size: 3.7" x 6.1"

7

ARCHITECTURAL HIGHLIGHTS

Complete RISC Computer

The IDT7RS104 is a complete reduced instruction set computer (RISC) module optimized for embedded control applications. It is based on the R3001 RISController and includes the R3010 Floating Point Accelerator (FPA) to enhance performance. On-board EPROM and serial I/O allow the module to perform as a stand-alone computer. All logic and control functions have been incorporated in the module in order to simplify system design for the end user. An 82C54 Programmable Interval Timer is included for periodic interrupt functions.

R3001 CPU

The module uses IDT's R3001 CPU device. The R3001 is architecturally the same as the R3000, but permits disabling of parity checking, thereby eliminating several bits of width in the synchronous memories. It is also capable of supporting synchronous memory spaces larger than the 256 KB maximum supported by the R3000. Like the R3000, the R3001 uses the R3010 Floating Point Accelerator for high speed arithmetic.

Simple Interface

For ease of use, all complex clock timing signals and handshaking logic are generated on the module and are derived from either an on-board oscillator, or an externally-supplied clock input signal. This frees the system designer from the task of having to design a state machine which implements the system's handshake and arbitration functions.

Large Synchronous Memory Space

A large "cache" memory is included on the module with 128K byte capacity for the Instruction Cache and 128K byte capacity for the Data Cache. This large cache allows most embedded control software to run completely within the cache, making external asynchronous memory unnecessary.

In fact, the 7RS104 "cache" memories are intended to be the machine's main memory. They are initially filled with instructions by using code in the EPROMs to move data from module's bus into memory using Load and Store instructions.

Reset and Initialization Logic

A master reset input triggers special initialization logic which is used to set the modes of operation of the R3001 with no user intervention required. The module is reset to Bigendian.

Two Synchronized User Interrupts

There are two pins for user interrupts provided on the module. They feed an on-board locked register to ensure synchronized activity with the R3000. The remaining four R3000 interrupts are used on the board: two interrupts are dedicated to the 82C54 Timer, one interrupt is dedicated to the 2681 DUART, and another interrupt is connected to the R3010 Floating Point Accelerator.

TYPICAL APPLICATIONS

The 7RS104 Module is designed to be a complete R3000 based controller, with a simple asynchronous handshaking to external I/O devices or possibly additional memory. The two 128 KB synchronous memory blocks on the module are intended to be large enough to store all the instructions and data used by the machine, so there are no cache miss stalls. This design not only allows the system to run at maximum performance, but also eliminates the inconsistencies in execution speed that result from cache misses and refills in conventional R3000 designs.

SIGNALS PROVIDED ON MODULE PINS

Pin Name	Type	Functional Description
D00-D31	I/O	Memory data lines to/from main memory system.
DP0-DP3	I/O	Parity bits for data lines. Parity must be supplied to the 7RS103 module, in accordance with R3000 requirements.
A00-A15	OUT	Address lines from the R3000. (lower 16 bits)
T16-T31	OUT	Tag lines from the R3000. (higher 16 bits)
ACCTYP0-	OUT	Positive-true outputs indicating the states of the R3000 ACCTYP2outputs, ACCTYP(0:2), which identify the nature of the data transactions for read/write cycles.
TAGV	OUT	Connection between cache and R3000.
INT0-INT5	IN	Interrupt inputs. Each has a 10K ohm pullup load resistor.
EXTOSC	IN	External oscillator input (Needed when using on-board delay line)
ECLKHPI		
ECLKRD SMP		
ECLKSYS	IN	The 2x Clock inputs for R3000 and R3010. Timings must conform to R3000 specifications. (Needed when not using on board delay line)
MRES#	IN	Negative-true master reset input. Connects directly to R3000 RES# input pin.
BSYSOUTA-	OUT	Buffered clock outputs for synchronizing external events.
BSYSOUTD		Each of these is an identical clock signal, representing the inverted form of the R3000 output, SYSOUT#.
BSYSOUT#	OUT	Buffered R3000 clock output, SYSOUT# for synchronizing external events. Non-inverted form of SYSOUT#.
MEMRD#	OUT	Direct negative-true output from R3000, used to indicate that a memory read cycle is in progress.
MEMWR#	OUT	Direct negative-true output from R3000, used to indicate that a memory write cycle is in progress.
RBSY	IN	Positive-true input used to request a memory read stall initiation and termination. This signal is normally held in its asserted state and deasserted at the completion of stalls.
WBSY#	IN	Negative-true input used to request a busy indication for subsequent memory write operations.
CPC0	IN	Direct input to the R3000, used to indicate the size of the data (block, word, byte, or other) for memory read cycles.
CPC1	OUT	Connection between the R3000 Processor and the R3010 FPA, indicating the status of the conditional branch. This pin is provided for diagnostic purposes, only.
CPC2, CPC3	I/O	Direct connections to R3000 pins.
EXC#	OUT	Direct output from R3000, indicating the EXC# signal between the R3000 and the R3010.
RUN#	OUT	Negative-true output from the R3000, indicating that the R3000 is in its RUN state (not stalled).
BERR#	IN	Negative-true input to R3000, used to indicate a bus error in main memory.
FPA#	OUT	Negative-true output indicating the presence of R3010 FPA on the module.
XEN#	OUT	Direct negative true output from the R3000. Used for read buffers output enable.
FPINT#	OUT	Negative-true R3010 interrupt request.

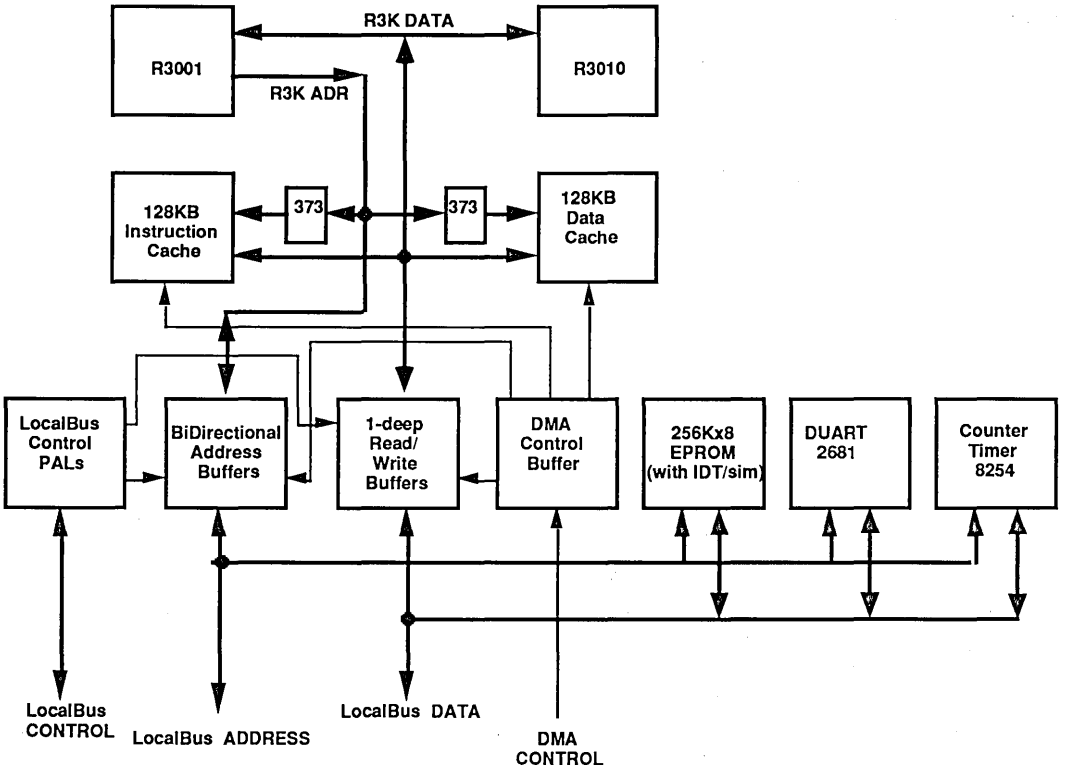
ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed
7RS103N44A16A	R3000A	NONE	16K	16K	16 MHz
7RS103N44A20A	R3000A	NONE	16K	16K	20 MHz
7RS103N44A25A	R3000A	NONE	16K	16K	25 MHz
7RS103F44A16A	R3000A	R3010A	16K	16K	16 MHz
7RS103F44A20A	R3000A	R3010A	16K	16K	20 MHz
7RS103F44A25A	R3000A	R3010A	16K	16K	25 MHz
7RS103N55A16A	R3000A	NONE	32K	32K	16 MHz
7RS103N55A20A	R3000A	NONE	32K	32K	20 MHz
7RS103N55A25A	R3000A	NONE	32K	32K	25 MHz
7RS103F55A16A	R3000A	R3010A	32K	32K	16 MHz
7RS103F55A20A	R3000A	R3010A	32K	32K	20 MHz
7RS103F55A25A	R3000A	R3010A	32K	32K	25 MHz
7RS103N66A16A	R3000A	NONE	64K	64K	16 MHz
7RS103N66A20A	R3000A	NONE	64K	64K	20 MHz
7RS103N66A25A	R3000A	NONE	64K	64K	25 MHz
7RS103N66A16A	R3000A	R3010A	64K	64K	16 MHz
7RS103N66A20A	R3000A	R3010A	64K	64K	20 MHz
7RS103N66A25A	R3000A	R3010A	64K	64K	25 MHz

MORE INFORMATION

For more information on this module, ask your IDT sales office for the Technical Specification and User's Manual.

BLOCK DIAGRAM



SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
STANDARD LocalBus SIGNALS		
MD00-MD31	I/O	Memory data lines to/from main memory system.
MA00-MA31	I/O	Memory address lines to main memory system. These are registered inputs/outputs.
LRD#	OUT	Negative-true output which indicates that a memory read cycle is in progress.
LWR#	OUT	Negative-true output which indicates that a memory write cycle is in progress.
RACK#	IN	Negative-true input which is used to indicate that the main memory read cycle initiated by the IDT7RS105 has been completed.
WACK#	IN	Negative-true input which is used to indicate that the main memory write cycle initiated by the IDT7RS105 has been completed.
BE0#-BE3#	OUT	Negative-true output which indicates which byte is being accessed during main memory read or write operations. These four signals are valid only when LRD# or LWR# istrue.
UINT0	IN	User interrupt input. Has a 10K ohm pullup load resistor.
UINT1	IN	Interrupt input which is driven by the R3010 Floating Point Accelerator on the standard module. Has a 10K ohm pullup load resistor.
UINT2	IN	User interrupt input. Has a 10K ohm pullup load resistor.
UINT3	IN	Interrupt input which is driven by the 8254 Counter/Timer on the standard module. Has a 10K ohm pullup load resistor.
UINT4	IN	Interrupt input which is driven by the 8254 Counter/Timer on the standard module. Has a 10K ohm pullup load resistor.
UINT5	IN	Interrupt input which is driven by the SN2681 DUART on the standard module. Has a 10K ohm pullup load resistor.
R3KOSCIN	IN	Oscillator input clock signal for the R3000/R3010. (2x clock rate).
BSYSOUT#	OUT	Buffered clock outputs for synchronizing external events. This signal represents the non-inverted form of the R3000 output, SYSOUT#.
BSYSCLK A-D	OUT	Buffered clock outputs for synchronizing external events. Each of these four outputs is an identical signal, representing the inverted form of the R3000 output SUSOUT#.
MREQ#	OUT	Negative-true, one-clock cycle long output which indicates the start of a main memory read or write cycle.
RESET#	IN	Negative-true master reset input.
CPT0	IN	Indirect input to R3000 Processor. Transfers conditional branch status from an external coprocessor to the R3000.
CPC1	OUT	Connection between the R3000 Processor and the R3010 FPA. Transfers conditional branch status from the R3010 to the R3000. This pin is provided for diagnostic purposes only.
CPC2-CPC3	IN	Direct inputs to the R3000 Processor. Transfers conditional branch status from external coprocessors to the R3000.
NON-STANDARD LocalBus SIGNALS		
UARTCLKOUT	OUT	Clock output from a 3.6864 MHz oscillator on board the module.
CLK8254	IN	Clock input to the 82C54 Interval Timer. May be shorted to UARTCLKOUT by the user.
EXTCSUART#	IN	Negative-true input which is the chip select for the SN2681 DUART.

List continued on following page

SIGNALS PROVIDED ON MODULE PINS (CONTINUED)

EXTCSTIM#	IN	Negative-true input which is the chip select for the 8254 Counter/ Timer.
INTCSUART#	OUT	Negative-true output of an on-board decoder. Decodes the UART address space as 0x1FE00000-0x1FE0FFFF. May be shorted to EXTCSUART# by the user.
INTCSTIM#	OUT	Negative-true output of an on-board decoder. Decodes the Interval Timer address space as 0x1FE40000-0x1FE4FFFF. May be shorted to EXTCSTIM# by the user.
DIRECT MEMORY ACCESS (DMA) CONTROL SIGNALS		
DICLK	IN	Latch enable to the instruction cache's address latch which is used during DMAs.
DDCLK	IN	Latch enable to the data cache's address latch which is used during DMAs.
DIWR#	IN	Negative-true input which is the write enable to the instruction cache RAMs. DIWR# is used during DMAs.
DDWR#	IN	Negative-true input which is the write enable to the data cache RAMs. DDWR# is used during DMAs.
DIRD#	IN	Negative-true input which is the output enable to the instruction cache RAMs. DIRD# is used during DMAs.
DDRD#	IN	Negative-true input which is the output enable to the data cache RAMs. DDRD# is used during DMAs.
TAGV	IN	Tag Valid input to the R3001. Has a 10K ohm pullup load resistor.
UDMA	IN	Positive-true, buffered input to the R3001's DMA pin. Has a 10K ohm pulldown load resistor.
DCTL#	IN	Negative-true, buffered input which is the output enable for all of the DMA signals. Can be driven true only after UDMA becomes true. Also acts as the output enable for the R3001 side of the address buffers. Has a 10K ohm pullup load resistor.
DAOE#	IN	Negative-true, buffered input which is the output enable to the main memory side of the address buffers. Can be used only when DCTL# is true. Has a 10K ohm pullupload resistor.
DMAEN#	IN	Negative-true, buffered input which is the clock enable to the main memory side of the address buffers. Can be used only when DCTL# is true.
DAEN#	IN	Negative-true, buffered input which is the clock enable to the R3001 side of the address buffers. Also act as the read data clock enable to the data buffers. Can only be used when DCTL# is true. Has a 10K ohm pullup load resistor.
DWOE#	IN	Negatvie-true, buffered input which is the write data output enable to the data buffers. Can only be used when DCTL# is true. Has a 10K ohm pulldown load resistor.
DXEN#	IN	Negative-true, buffered input which is the read data output enable to the data buffers. Can only be used when DCTL# is true. Has a 10K ohm pullup load resistor.
DUART CONTROL SIGNALS		
RxDA	IN	Direct SCN2681 input.
RxDB	IN	Direct SCN2681 input.
TxDA	OUT	Direct SCN2681 output.
TxDB	OUT	Direct SCN2681 output.
IP0-IP1	IN	Direct SCN2681 inputs.
IP2-IP6		
OP0-OP7	OUT	Direct SCN2681 outputs.

RELATED PRODUCTS

IDT/sim

The 7RS104 module includes IDT's monitor in EPROM on board. The IDT7RS901 System Integration Manager (IDT/sim) is a ROMable software product that permits convenient control and debug of RISC systems built around the MIPS R3000 architecture. It permits users to quickly develop and debug stand-alone systems. Facilities are included to operate the CPU under controlled conditions, examining and altering the contents of memory, manipulating and controlling R3000 resources (such as cache, TLB and coprocessors), loading programs from host machines, and controlling the path of execution of loaded programs. Remote (source/symbolic) debugging is also supported.
IDT/sim requires 82Kb of EPROM space for code and data and 16Kb of RAM space for uninitialized variable data and stack. The minimal I/O system supported uses UARTS. The default drivers support the 2681 or 68681 devices. Other devices can be added easily.

Prototyping Platform

A Prototyping Platform is in development for this product. Please contact your IDT sales office for latest status and technical information.

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-memory	D-memory	Speed	Other
7RS104F77A16A	R3001	R3010	128KB	128KB	16 MHz	EPROM socketed
7RS104F77A20A	R3001	R3010	128KB	128KB	20 MHz	EPROM socketed
7RS104F77A25A	R3001	R3010	128KB	128KB	25 MHz	EPROM socketed

CUSTOM OPTIONS

Most software features of the 7RS104 can be modified by special order. Contact sales office for details.

MORE INFORMATION

For more information on this module, ask your IDT sales office for the Technical Specification and User's Manual.

Integrated Device Technology, Inc.

R3000 CPU MODULES FOR HIGH PERFORMANCE AND MULTIPROCESSOR SYSTEMS

IDT7RS107

FEATURES:

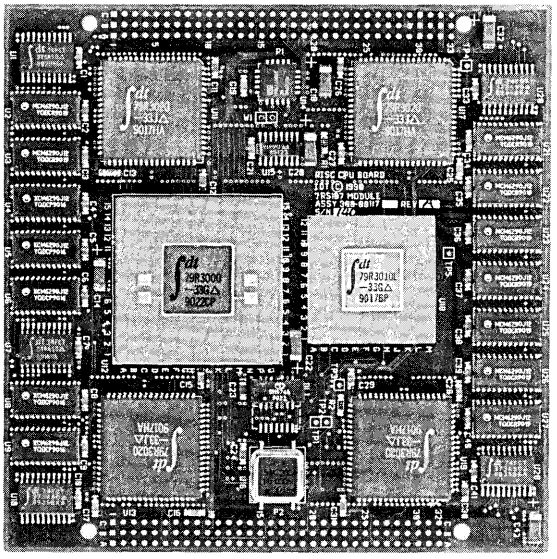
- Cache Size: 64K Instruction, 64K Data
- Processor Speeds up to 33 MHz
- Includes R3010 Floating Point Accelerator
- 1-word Read Buffer; 4-word Write Buffer
- Supports R3000 Multiprocessor Features
- Entire I-Cache can be invalidated with external cache reset signal
- Eight-word block refills
- On-board oscillator, delay line, and reset circuitry
- 100% burn-in and functional test at rated speed

R3000 MODULE FOR HIGH PERFORMANCE CPUs AND MULTIPROCESSOR SYSTEMS:

The IDT7RS107 is a complete reduced instruction set computer (RISC) CPU, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the R3000 CPU, the R3010 Floating Point Accelerator, 64 Kbytes each of data and instruction cache memory, a single word read buffer and a four-word write buffer. Clock generation, reset, control and interrupt functions are included on the module to simplify the remainder of the system design.

The 107 module is designed to support the R3000's multiprocessor features. Data in the D-cache can be invalidated by the R3000 CPU. It is also possible to invalidate the entire contents of the I-cache in a single cycle by using an external cache reset signal.

The module is constructed using surface mount devices on a 5.2" by 5.2" epoxy laminate board, and is connected to the user's system via 195 pins located in two pin row regions on the board.



7RS107 Module. Actual Size 5.2" x 5.2 "

7

ARCHITECTURAL HIGHLIGHTS

Uses R3020 Write Buffers

R3020 chips are used on the module to provide a "smart" four-deep write buffer between the CPU and external memory. These devices store data and addresses for up to four write requests to main memory, and handle the handshaking with the memory controller. The R3020s support features such as byte gathering (combining multiple byte writes to the same address in the buffer into a single write) and address matching (a read or write to an address already in the write buffer will be detected so the user software can take appropriate action). The R3020's Match signals are OR'ed on the module to produce a single output, labeled CONFLICT.

Resettable Instruction Cache

The 7RS107 module permits invalidation of the entire instruction cache via a "cache reset" pin on the module. This feature is used to wipe the cache clean when the a block of instructions in main memory have been changed by a DMA operation. It is usually much faster than invalidating each affecting tag individually.

Multiprocessor Invalidate In Data Cache

The module supports the R3000's multiprocessor cache invalidate feature, so that data cache coherency can be maintained when data held in the cache is altered externally. The R3000's MP Stall and MP Invalidate signals are available as pins on the module. The user's system stalls the processor and then provides an address to the module while signaling MP Invalidate. The module stores the address in a latch and applies it to the cache at the right time for the R3000 to invalidate the referenced tag.

Eight-Word Block Refill

The module refills both the instruction and data caches from memory in eight-word blocks. Following a cache miss, the processor will request a memory read at the missed address and wait for a data ready acknowledgement. When an acknowledge is received, the processor will load eight words into cache on eight successive clock cycles. The memory interface must supply the correct eight words (address A4A3A2 = 0 to 7) at the processor's speed, 40 ns intervals for a 25 MHz system. Interleaved memory is usually the best way to support this requirement. The processor's CPC0 pin, available as a pin on the module, can be used to over-ride the block refill on data, but instructions refills must always be in 8-word blocks. The processor performs instruction streaming during the refill.

On-board Oscillator and Delay Line

All the clock generation circuitry required by the R3000 system is on the module. A jumper can be used to select between the on-board crystal oscillator or an external oscillator input. A delay line on the module is used to set the timing for register strobes and other critical signals relative to the R3000 clock. The R3000 clock output "SYSOUT" is made available to the user system through eight pins on the module, each independently buffered.

R3000 Reset and Initialization Logic

The initialization logic for the R3000 CPU is contained on the module. A "Cold Reset" pin on the module starts the required 15 ms reset signal to the CPU, and then provides the initialization vectors during the last few cycles. A second reset pin is provided to reinitialize the CPU without repeating the 15 ms delay. The R3000 is initialized to "Big-Endian" operation.

Five User Interrupt Lines

Five pins on the module are used for user interrupt inputs. The user interrupts are synchronized in registers on the module before being sent to the R3000. Interrupt 2 is used for the Floating Point Accelerator, if present.

External R3000 Condition Code Pin

The R3000 input CPC0 is available as a pin on the module. During initialization, this pin is programmed as a Condition Code test pin, so the R3000 can do a Test and Branch in a single cycle based on its state. During read stalls, the pin determines whether a single word or 8 words will be read. Reads into the instruction cache must always be block refills.

TYPICAL APPLICATIONS

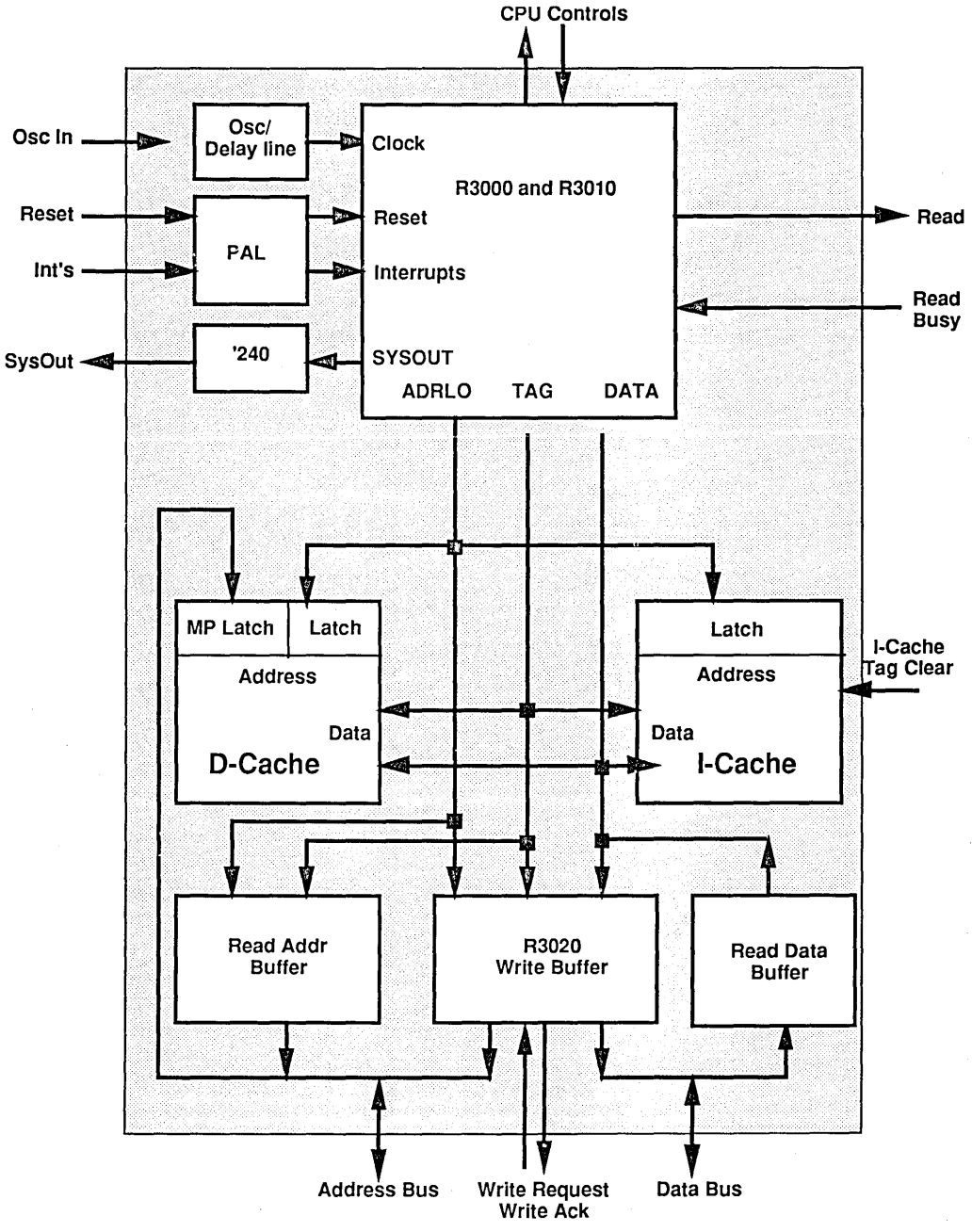
The 7RS107 module is designed for applications that run complex operating systems, such as UNIX™, or that need outside control of cache memory contents, such as multiprocessor systems.

The module supports the R3000's ability to invalidate entries in the data cache, allowing multiple processor systems to maintain cache coherency.

The module is offered with the maximum possible cache sizes (64K each) that can be supported by the R3000 in a multiprocessor configuration. These sizes are well suited to running UNIX at very high instruction rates as well.

The R3020 Write Buffer is used to provide a four-word deep write buffer, which is ideal for most UNIX systems.

FUNCTIONAL BLOCK DIAGRAM



7

SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
MA0...MA31	I/O	32-bit address from the module to external memory. This is an output from the 3020 Write Buffer except during the MP Invalidate function, when it is the input to the MP cache address latch.
MD0...MD31	I/O	32-bit data bus between the module and external memory. Driven from the 3020 Write Buffer during writes; input to the Read Data Buffer during reads.
BACT0,1,2	O	The three R3000 AccType status signals, driven from the 3020 Write Buffer during writes and from a latch during reads.
MDP0...MDP3	I/O	The four parity bits for the MD data. Output during writes and input during reads.
CP_CpCond0, 2, 3	I	The three flag inputs to the R3000 CPU. CPC0 is used during read stalls to control block refill of the data cache. (The instruction cache must always be block refilled.) CPC2 and CPC3 are the MP stall and invalidate controls.
ALOE	I	Data Cache Address Latch Output Enable When LOW, enables the output of the latch holding the data cache address supplied by the R3000. It should be LOW at all times except when the MP Latch is being used to invalidate a cache address.
MPALOE	I	Data Cache MP Address Latch Output Enable. This input is used to enable the output of the latch holding the address supplied by the user system during an MP stall cycle. It should be enabled (LOW) only during the MP invalidate operation.
BSYSOUT2...9	O	Eight buffered inverted copies of the R3000 signal "SYSOUT" for use in the user's system.
UINT0,1,3,4,5	I	Interrupt inputs to the R3000. These signals are synchronized to SYSOUT on the module. R3000 interrupt 2 is used for the Floating Point Accelerator.
BRESET	O	Buffered copy of the reset signal created on the module to reset the CPU. LOW during Reset.
WB_WbFull	O	Write Busy. Status signal created by the R3020 write buffer. Goes LOW to indicate the buffer is full.
CPU_BusError	I	Input to the R3000 indicating a bus error has occurred.
RESETC	I	Cold Reset to the module. The module creates a 15 ms long reset to the R3000 and executes the R3000 initialization sequence when this pin goes LOW.
FP_FpPresent	O	This signal can be used to detect the presence of an FPA on the module. To be used, it must be connected to a 4.7K pullup resistor. The pin will be LOW if the FPA is present.
RESETI	I	Active LOW asynchronous clear to the I-Cache Tag RAMS. Sets the entire I-Cache invalid.
WB_OutEn	I	Write Buffer Output Enable. When LOW, turns on the outputs of the R3020 write buffers.
WB_Request	O	Output from the R3020 to indicate that there is data in the buffer to be written to memory. Active LOW
WB_Acknowledge	I	Input to the R3020 to indicate data has been written into memory.
CONFLICT	O	The OR of all the R3020 Match signals; indicates the address on the R3020 inputs matches one of the addresses currently in the write buffer.
RABOE	I	Read Address Buffer Output Enable. When LOW, turns on outputs of the buffers containing the read address.
RDBCE	I	Read Data Buffer Clock Enable. When LOW, enables the clock (SYSOUT) to the Read Data Buffers.
READ	O	Status signal output. LOW during reads.
RABLE	I	Read Address Buffer Latch Enable. When HIGH, enables the Read Address Buffer latches.
WB_LatchErrAddr	I	Latch Error Address input to the R3020.
WB_EnErrAddr	I	Enable Error Address input to R3020.
CP_MemRd	O	R3000 output signal. When LOW, there is a request for a read from external memory.
CP_RdBusy	I	Read Busy. Input to the R3000 to indicate acknowledgment of the MEMRD request.
RESETX	I	Additional Reset command. Same as RESETC, but does not go through the 15 ms delay. Can be used to re-initialize the R3000 when power is on.

RELATED PRODUCTS

Prototyping System

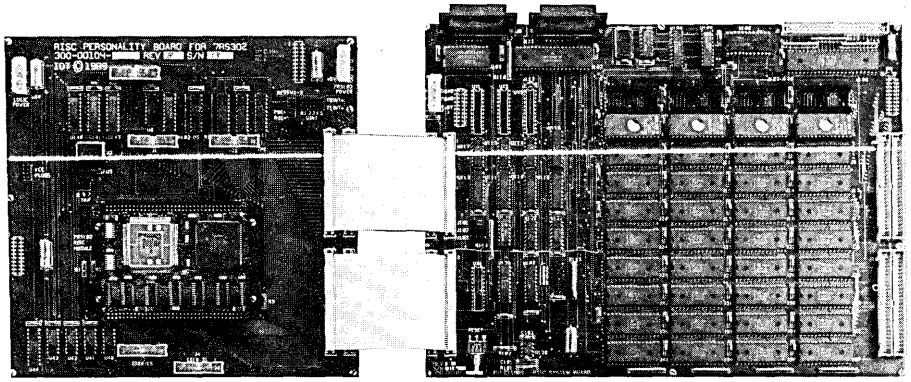
The 7RS107 module can be placed into immediate service using our flexible 7RS307 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's Software Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's PC-based cross assembler and compiler products. Assembled code can be downloaded into the Prototyping System for execution and debug.



A Module Prototyping Platform.
The card on the left is the personality card with a module; the card on the right is the general purpose CPU.

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed	Other
7RS107N66A16A	R3000A	NONE	64K	64K	16 MHz	
7RS107N66A20A	R3000A	NONE	64K	64K	20 MHz	
7RS107N66A25A	R3000A	NONE	64K	64K	25 MHz	
7RS107N66A33A	R3000A	NONE	64K	64K	33 MHz	
7RS107F66A16A	R3000A	R3010A	64K	64K	16 MHz	
7RS107F66A20A	R3000A	R3010A	64K	64K	20 MHz	
7RS107F66A25A	R3000A	R3010A	64K	64K	25 MHz	
7RS107F66A33A	R3000A	R3010A	64K	64K	33 MHz	

CUSTOM OPTIONS

Some features of the 7RS107 can be modified by special order. Contact your IDT sales office for details.

Software modifications include: initialization mode for the R3000, endian option, size of block refill, instruction streaming option.

Manufacturing options include pin length, style, and plating; special marking; additional burn-in, and socketing of the CPU and/or FPA.

Integrated Device Technology, Inc.

R3000 CPU MODULES WITH 256K CACHES

IDT7RS108

FEATURES:

- **Cache Size: 256KB Instruction, 256KB Data**
- **Processor Speeds up to 25 MHz**
- **Includes R3010 Floating Point Accelerator**
- **1-word Read Buffer; 4-word Write Buffer**
- **Eight-word block refills**
- **On-board Parity Generation and Check**
- **On-board oscillator, delay line, and reset circuitry**
- **100% burn-in and functional test at rated speed**

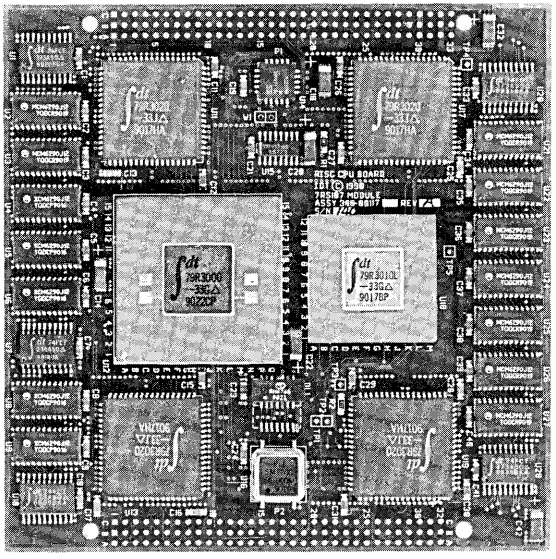
R3000 MODULE FOR HIGH PERFORMANCE CPUs:

The IDT7RS108 is a complete reduced instruction set computer (RISC) CPU, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the R3000 CPU, the R3010 Floating Point Accelerator, 256 Kbytes each of data and instruction cache memory, a single word read buffer and a four-word write buffer.

Clock generation, reset, parity, control and interrupt functions are included on the module to simplify the remainder of the system design.

The 7RS108 module is pin compatible with the 7RS107 and 7RS109 modules (with 64K caches), but does not support the multiprocessing features offered by those modules.

The module is constructed using surface mount devices on a 5.2" by 5.2" epoxy laminate board, and is connected to the user's system via 195 pins located in two pin row regions on the board.



7RS108 Module. Actual Size 5.2" x 5.2 "

7

ARCHITECTURAL HIGHLIGHTS

Uses R3020 Write Buffers

R3020 chips are used on the module to provide a "smart" four-deep write buffer between the CPU and external memory. These devices store data and addresses for up to four write requests to main memory, and handle the handshaking with the memory controller. The R3020s support features such as byte gathering (combining multiple byte writes to the same address in the buffer into a single write) and address matching (a read or write to an address already in the write buffer will be detected so the user software can take appropriate action). The R3020's Match signals are OR'ed on the module to produce a single output, labeled CONFLICT.

Eight-Word Block Refill

The module refills both the instruction and data caches from memory in eight-word blocks. Following a cache miss, the processor will request a memory read at the missed address and wait for a data ready acknowledgement. When an acknowledge is received, the processor will load eight words into cache on eight successive clock cycles. The memory interface must supply the correct eight words (address A4A3A2 = 0 to 7) at the processor's speed, 40 ns intervals for a 25 MHz system. Interleaved memory is usually the best way to support this requirement. The processor's CPC0 pin, available as a pin on the module, can be used to over-ride the block refill. The processor performs instruction streaming during the refill.

On-board Oscillator and Delay Line

All the clock generation circuitry required by the R3000 system is on the module. A jumper can be used to select between the on-board crystal oscillator or an external oscillator input. A delay line on the module is used to set the timing for register strobes and other critical signals relative to the R3000 clock. The R3000 clock output "SYSOUT" is made available to the user system through eight pins on the module, each independently buffered.

R3000 Reset and Initialization Logic

The initialization logic for the R3000 CPU is contained on the module. A "Cold Reset" pin on the module starts the required 15 ms reset signal to the CPU, and then provides the initialization vectors during the last few cycles. A second reset pin is provided to reinitialize the CPU without repeating the 15 ms delay. The R3000 is initialized to "Big-Endian" operation.

Five User Interrupt Lines

Five pins on the module are used for user interrupt inputs. The user interrupts are synchronized in registers on the module before being sent to the R3000. Interrupt 2 is used for the Floating Point Accelerator, if present.

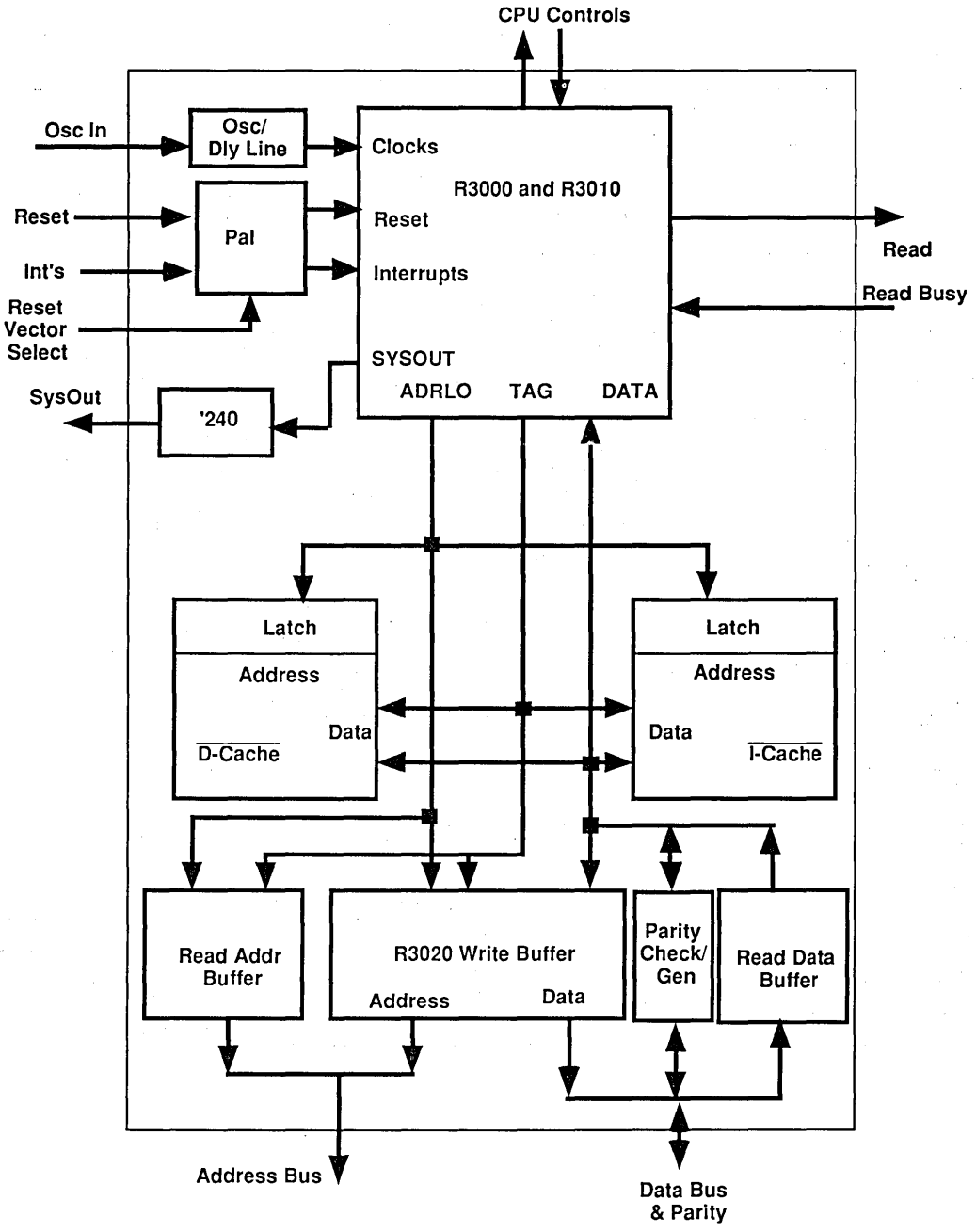
External R3000 Condition Code Pin

The R3000 input CPC0 is available as a pin on the module. During initialization, this pin is programmed as a Condition Code test pin, so the R3000 can do a Test and Branch in a single cycle based on its state. During read stalls, the pin determines whether a single word or 8 words will be read. Reads into the instruction cache must always be block refills.

TYPICAL APPLICATIONS

The 7RS108 module is designed for applications that run complex operating systems, such as UNIX™, or that need the maximum possible performance. The module contains the maximum possible cache sizes (256K each) that can be supported by the R3000.

FUNCTIONAL BLOCK DIAGRAM



SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
MA0...MA31	I/O	32-bit address from the module to external memory. This is an output from the 3020 Write Buffer except during the MP Invalidate function, when it is the input to the MP cache address latch.
MD0...MD31	I/O	32-bit data bus between the module and external memory. Driven from the 3020 Write Buffer during writes; input to the Read Data Buffer during reads.
BACT0,1,2	O	The three R3000 AccType status signals, driven from the 3020 Write Buffer during writes and from a latch during reads.
MDP0...MDP3	I/O	The four parity bits for the MD data. Output during writes and input during reads.
CP_CpCond0, 2, 3	I	The three flag inputs to the R3000 CPU. CPC0 is used during read stalls to control block refill of the data cache. (The instruction cache must always be block refilled.) CPC2 and CPC3 are the MP stall and invalidate controls.
ALOE	I	Data Cache Address Latch Output Enable When LOW, enables the output of the latch holding the data cache address supplied by the R3000. It should be LOW at all times except when the MP Latch is being used to invalidate a cache address.
BSYSOUT2...9	O	Eight buffered inverted copies of the R3000 signal "SYSOUT" for use in the user's system.
UINT0,1,3,4,5	I	Interrupt inputs to the R3000. These signals are synchronized to SYSOUT on the module. R3000 interrupt 2 is used for the Floating Point Accelerator.
BRESET	O	Buffered copy of the reset signal created on the module to reset the CPU. LOW during Reset.
WB_WbFull	O	Write Busy. Status signal created by the R3020 write buffer. Goes LOW to indicate the buffer is full.
CPU_BusError	I	Input to the R3000 indicating a bus error has occurred.
RESETC	I	Cold Reset to the module. The module creates a 15 ms long reset to the R3000 and executes the R3000 initialization sequence when this pin goes LOW.
FP_FpPresent	O	This signal can be used to detect the presence of an FPA on the module. To be used, it must be connected to a 4.7K pullup resistor. The pin will be LOW if the FPA is present.
WB_OutEn	I	Write Buffer Output Enable. When LOW, turns on the outputs of the R3020 write buffers.
WB_Request	O	Output from the R3020 to indicate that there is data in the buffer to be written to memory. Active LOW
WB_Acknowledge	I	Input to the R3020 to indicate data has been written into memory.
CONFLICT	O	The OR of all the R3020 Match signals; indicates the address on the R3020 inputs matches one of the addresses currently in the write buffer.
RABOE	I	Read Address Buffer Output Enable. When LOW, turns on outputs of the buffers containing the read address.
RDBCE	I	Read Data Buffer Clock Enable. When LOW, enables the clock (SYSOUT) to the Read Data Buffers.
READ	O	Status signal output. LOW during reads.
RABLE	I	Read Address Buffer Latch Enable. When HIGH, enables the Read Address Buffer latches.
WB_LatchErrAddr	I	Latch Error Address input to the R3020.
WB_EnErrAddr	I	Enable Error Address input to R3020.
CP_MemRd	O	R3000 output signal. When LOW, there is a request for a read from external memory.
CP_RdBusy	I	Read Busy. Input to the R3000 to indicate acknowledgment of the MEMRD request.

RELATED PRODUCTS

Prototyping System

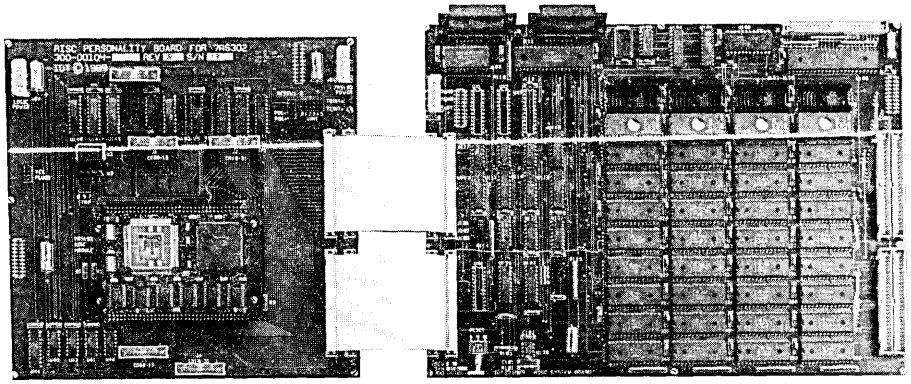
The 7RS108 module can be placed into immediate service using our flexible 7RS308 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's Software Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's PC-based cross assembler and compiler products. Assembled code can be downloaded into the Prototyping System for execution and debug.



A Module Prototyping Platform.
The card on the left is the personality card with a module; the card on the right is the general purpose CPU.

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-Cache	D-Cache	Speed	Other
7RS108N88A16A	R3000A	NONE	256K	256K	16 MHz	
7RS108N88A20A	R3000A	NONE	256K	256K	20 MHz	
7RS107N88A25A	R3000A	NONE	256K	256K	25 MHz	
7RS108F88A16A	R3000A	R3010A	256K	256K	16 MHz	
7RS108F88A20A	R3000A	R3010A	256K	256K	20 MHz	
7RS108F88A25A	R3000A	R3010A	256K	256K	25 MHz	

For detailed design information, contact IDT and ask for the 7RS108 Technical Specification.

CUSTOM OPTIONS

Some features of the 7RS108 can be modified by special order. Contact your IDT sales office for details.

Software modifications include: initialization mode for the R3000, endian option, size of block refill, instruction streaming option.

Manufacturing options include pin length, style, and plating; special marking; additional burn-in, and socketing of the CPU and/or FPA.

Integrated Device Technology, Inc.

R3000 CPU MODULES

ADVANCE INFORMATION
IDT7RS109

FEATURES:

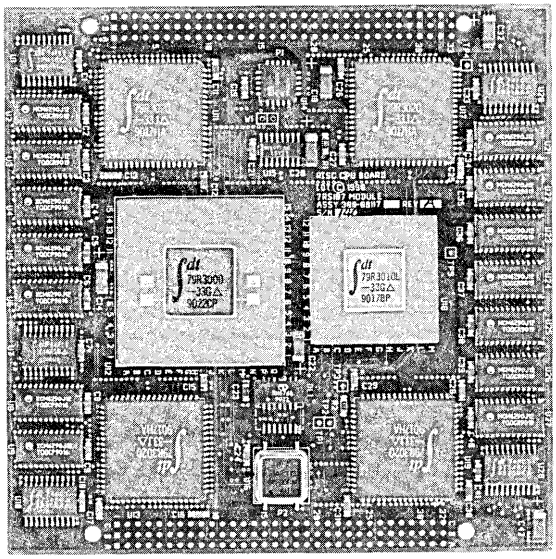
- Cache Size: 64K Instruction, 64K Data
- Processor Speeds up to 33 MHz
- Includes R3010 Floating Point Accelerator
- 1-word Read Buffer; 4-word Write Buffer
- Supports R3000 Multiprocessor Features
- On-Board Parity Check and Generate
- Eight-word block refills
- On-board oscillator, delay line, and reset circuitry
- 100% burn-in and functional test at rated speed

R3000 MODULE FOR HIGH PERFORMANCE CPUs AND MULTIPROCESSOR SYSTEMS:

The IDT7RS109 is a complete reduced instruction set computer (RISC) CPU, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. The module includes the R3000 CPU, the R3010 Floating Point Accelerator, 64 Kbytes each of data and instruction cache memory, a single word read buffer and a four-word write buffer. Clock generation, reset, control, parity, and interrupt functions are included on the module to simplify the remainder of the system design.

The 109 module includes a latch to hold an external address for snooping in the D-cache and is designed to support the R3000's multiprocessor features.

The module is constructed using surface mount devices on a 5.2" by 5.2" epoxy laminate board, and is connected to the user's system via 195 pins located in two pin row regions on the board.



7RS109 Module. Actual Size 5.2" x 5.2 "

7

ARCHITECTURAL HIGHLIGHTS

Uses R3020 Write Buffers

R3020 chips are used on the module to provide a "smart" four-deep write buffer between the CPU and external memory. These devices store data and addresses for up to four write requests to main memory, and handle the handshaking with the memory controller. The R3020s support features such as byte gathering (combining multiple byte writes to the same address in the buffer into a single write) and address matching (a read or write to an address already in the write buffer will be detected so the user software can take appropriate action). The R3020's Match signals are OR'ed on the module to produce a single output, labeled CONFLICT.

Eight-Word Block Refill

The module refills both the instruction and data caches from memory in eight-word blocks. Following a cache miss, the processor will request a memory read at the missed address and wait for a data ready acknowledgement. When an acknowledge is received, the processor will load eight words into cache on eight successive clock cycles. The memory interface must supply the correct eight words (address A4A3A2 = 0 to 7) at the processor's speed, 40 ns intervals for a 25 MHz system. Interleaved memory is usually the best way to support this requirement. The processor's CPC0 pin, available as a pin on the module, can be used to over-ride the block refill. The processor performs instruction streaming during the refill.

On-board Oscillator and Delay Line

All the clock generation circuitry required by the R3000 system is on the module. A jumper can be used to select between the on-board crystal oscillator or an external oscillator input. A delay line on the module is used to set the timing for register strobes and other critical signals relative to the R3000 clock. The R3000 clock output "SYSOUT" is made available to the user system through eight pins on the module, each independently buffered.

R3000 Reset and Initialization Logic

The initialization logic for the R3000 CPU is contained on the module. A "Cold Reset" pin on the module starts the required 15 ms reset signal to the CPU, and then provides the initialization vectors during the last few cycles. A second reset pin is provided to reinitialize the CPU without repeating the 15 ms delay. The R3000 is initialized to "Big-Endian" operation.

Five User Interrupt Lines

Five pins on the module are used for user interrupt inputs. The user interrupts are synchronized in registers on the module before being sent to the R3000. Interrupt 2 is used for the Floating Point Accelerator, if present.

External R3000 Condition Code Pin

The R3000 input CPC0 is available as a pin on the module. During initialization, this pin is programmed as a Condition Code test pin, so the R3000 can do a Test and Branch in a single cycle based on its state. During read stalls, the pin determines whether a single word or 8 words will be read. Reads into the instruction cache must always be block refills.

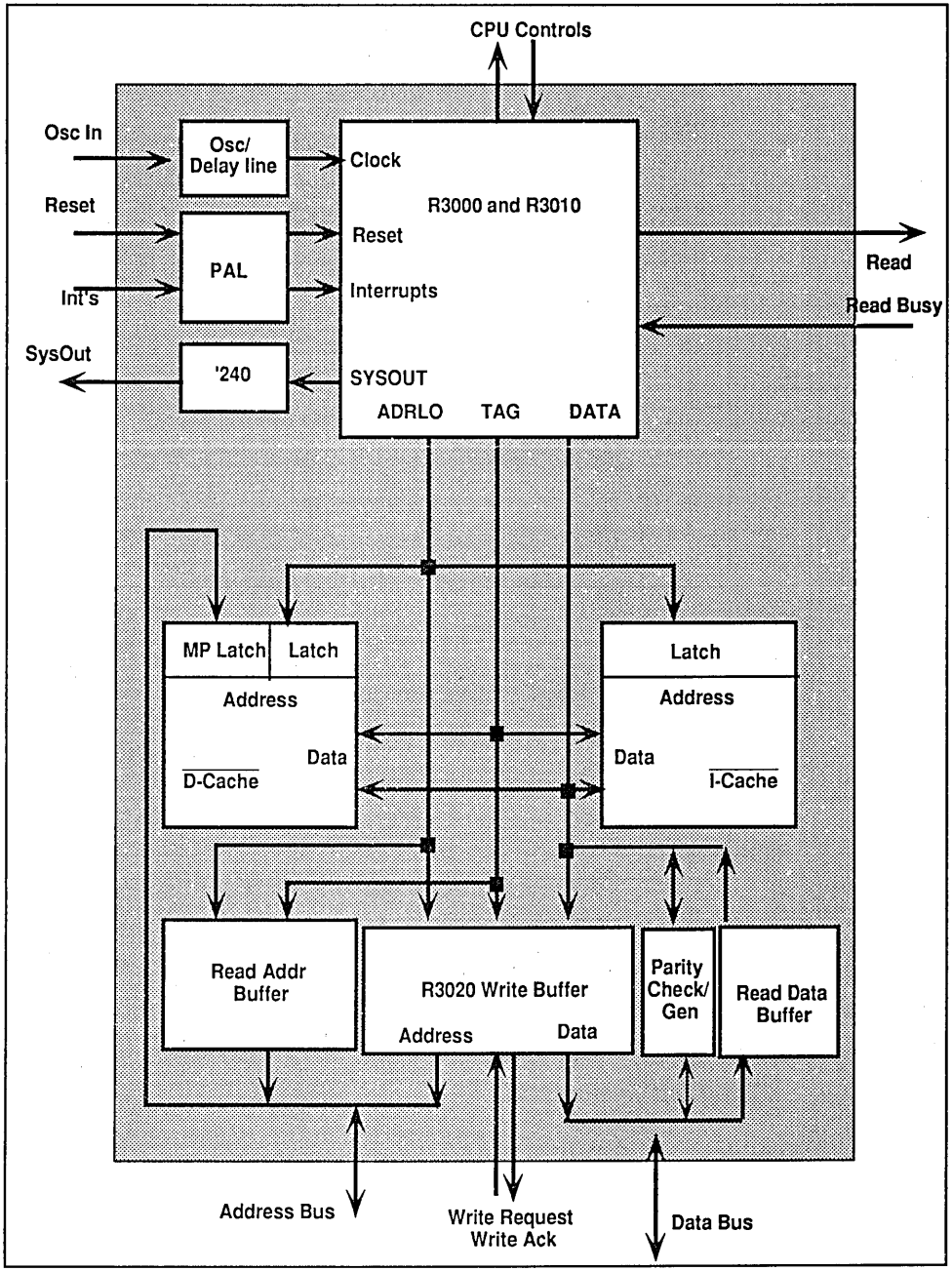
Internal Parity Check and Generate

The 7RS109 uses IDT 73211 registers for the data read buffer. This device provides the ability to generate parity on incoming data, if it is not already present, or to check parity on outgoing data to detect parity errors occuring on the module.

TYPICAL APPLICATIONS

The 7RS109 module is designed for applications that run complex operating systems, such as UNIX™, or that need the maximum possible performance. The module contains the maximum possible cache sizes (64K each) that can be supported by the R3000 in Multiprocessor configurations.

FUNCTIONAL BLOCK DIAGRAM



7

SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
MA0...MA31	I/O	32-bit address from the module to external memory. This is an output from the 3020 Write Buffer except during the MP Invalidate function, when it is the input to the MP cache address latch.
MD0...MD31	I/O	32-bit data bus between the module and external memory. Driven from the 3020 Write Buffer during writes; input to the Read Data Buffer during reads.
BACT0,1,2	O	The three R3000 AccType status signals, driven from the 3020 Write Buffer during writes and from a latch during reads.
MDP0...MDP3	I/O	The four parity bits for the MD data. Output during writes and input during reads.
CP_CpCond0, 2, 3	I	The three flag inputs to the R3000 CPU. CPC0 is used during read stalls to control block refill of the data cache. (The instruction cache must always be block refilled.) CPC2 and CPC3 are the MP stall and invalidate controls.
ALOE	I	Data Cache Address Latch Output Enable When LOW, enables the output of the latch holding the data cache address supplied by the R3000. It should be LOW at all times except when the MP Latch is being used to invalidate a cache address.
BSYSOUT2...9	O	Eight buffered inverted copies of the R3000 signal "SYSOUT" for use in the user's system.
UINT0,1,3,4,5	I	Interrupt inputs to the R3000. These signals are synchronized to SYSOUT on the module. R3000 interrupt 2 is used for the Floating Point Accelerator.
BRESET	O	Buffered copy of the reset signal created on the module to reset the CPU. LOW during Reset.
WB_WbFull	O	Write Busy. Status signal created by the R3020 write buffer. Goes LOW to indicate the buffer is full.
CPU_BusError	I	Input to the R3000 indicating a bus error has occurred.
RESETC	I	Cold Reset to the module. The module creates a 15 ms long reset to the R3000 and executes the R3000 initialization sequence when this pin goes LOW.
FP_FpPresent	O	This signal can be used to detect the presence of an FPA on the module. To be used, it must be connected to a 4.7K pullup resistor. The pin will be LOW if the FPA is present.
WB_OutEn	I	Write Buffer Output Enable. When LOW, turns on the outputs of the R3020 write buffers.
WB_Request	O	Output from the R3020 to indicate that there is data in the buffer to be written to memory. Active LOW
WB_Acknowledge	I	Input to the R3020 to indicate data has been written into memory.
CONFLICT	O	The OR of all the R3020 Match signals; indicates the address on the R3020 inputs matches one of the addresses currently in the write buffer.
RABOE	I	Read Address Buffer Output Enable. When LOW, turns on outputs of the buffers containing the read address.
RDBCE	I	Read Data Buffer Clock Enable. When LOW, enables the clock (SYSOUT) to the Read Data Buffers.
READ	O	Status signal output. LOW during reads.
RABLE	I	Read Address Buffer Latch Enable. When HIGH, enables the Read Address Buffer latches.
WB_LatchErrAddr	I	Latch Error Address input to the R3020.
WB_EnErrAddr	I	Enable Error Address input to R3020.
CP_MemRd	O	R3000 output signal. When LOW, there is a request for a read from external memory.
CP_RdBusy	I	Read Busy. Input to the R3000 to indicate acknowledgment of the MEMRD request.

RELATED PRODUCTS

Prototyping System

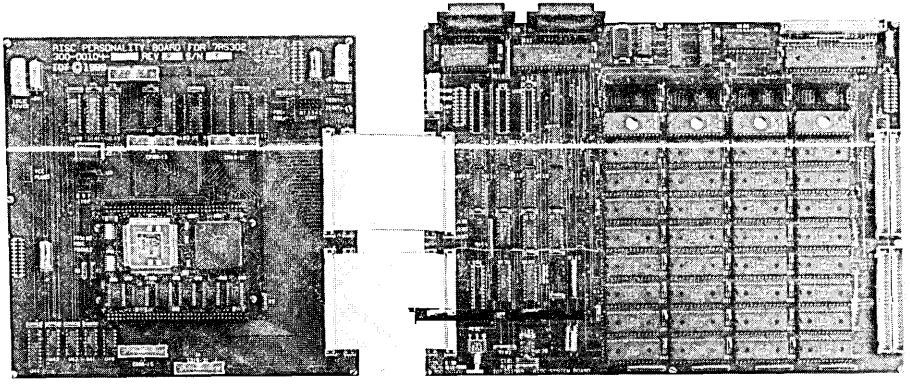
The 7RS109 module can be placed into immediate service using our flexible 7RS309 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's Software Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's PC-based cross assembler and compiler products. Assembled code can be downloaded into the Prototyping System for execution and debug.



A Module Prototyping Platform.
The card on the left is the personality card with a module; the card on the right is the general purpose CPU.

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed	Other
7RS109N66A16A	R3000A	NONE	64K	64K	16 MHz	
7RS109N66A20A	R3000A	NONE	64K	64K	20 MHz	
7RS107N66A25A	R3000A	NONE	64K	64K	25 MHz	
7RS107N66A33A	R3000A	NONE	64K	64K	33 MHz	
7RS109F66A16A	R3000A	R3010A	64K	64K	16 MHz	
7RS109F66A20A	R3000A	R3010A	64K	64K	20 MHz	
7RS109F66A25A	R3000A	R3010A	64K	64K	25 MHz	
7RS109F66A33A	R3000A	R3010A	64K	64K	33 MHz	

CUSTOM OPTIONS

Some features of the 7RS109 can be modified by special order. Contact your IDT sales office for details.

Software modifications include: initialization mode for the R3000, endian option, size of block refill, instruction streaming option.

Manufacturing options include pin length, style, and plating; special marking; additional burn-in, and socketing of the CPU and/or FPA.

Integrated Device Technology, Inc.

PLUG COMPATIBLE FAMILY OF R3000 CPU MODULES

IDT7RS110

FEATURES:

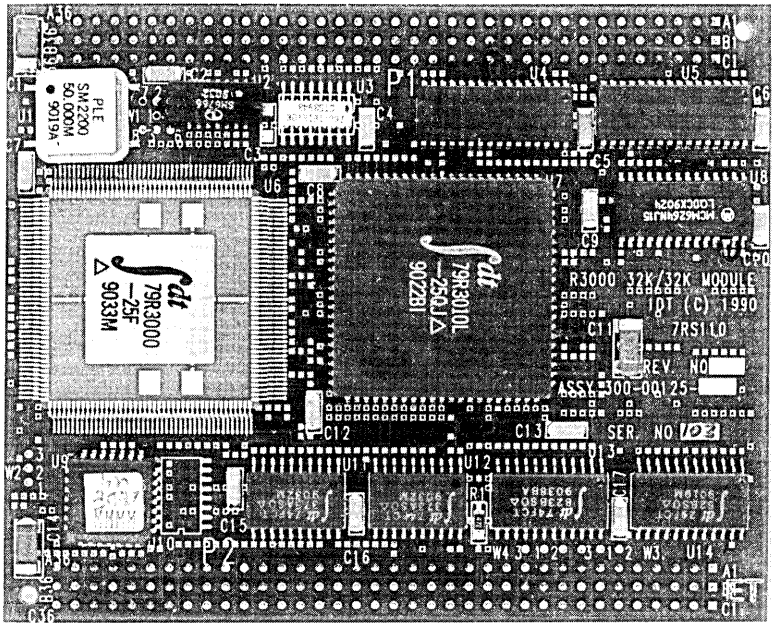
- **Choice of Cache Sizes: 32 I and D or 32K I and 16K D**
- **Small size: 2.9" x 4.1"**
- **Processor Speeds up to 33 MHz**
- **Optional R3010 Floating Point Accelerator**
- **Single word Read and Write Buffers**
- **On-Board Clock, Reset, and Parity Logic**
- **100% burn-in and functional test at rated speed**

R3000 MODULE FOR GENERAL USE IN SMALL SYSTEMS:

The IDT7RS110 is a complete reduced instruction set computer (RISC) SubSystem, based on the MIPS R3000 RISC processor, and supplied on a small fully-tested high-density plug-in module. There are two versions of the module. One version contains 32 Kbytes each of data and instruction cache memory, and the other 32 Kbytes of instruction and 16 Kbytes of data cache. Both include an R3000 Instruction Set CPU, a single word read buffer, a single word write buffer and, optionally, the R3010 Floating Point Accelerator.

Cache misses are handled with single word read requests to memory, providing a simple interface to any type of main memory system.

The module is constructed using surface mount devices on a two-sided epoxy laminate board, and is connected to the user's system via six rows of 36 pins each.



7RS110 Module. Actual Size 2.9" x 4.1 "

ARCHITECTURAL HIGHLIGHTS

Interchangeable Performance Options

The 7RS110 is available in two cache size choices and in a variety of speed selections. All versions plug into the same footprint, creating a simple way to offer price/performance selections for RISC systems.

R3010 Floating Point Accelerator

The R3010 Floating Point Accelerator (FPA) is included as an integral part of the module. It operates in conjunction with the R3000 RISC Processor and greatly improves the system performance by expanding the instruction set to include very fast floating point capabilities. All timing and control connections are on the module. The R3010 can use either INT1 or INT3 to the R3000 (jumper selectable).

On-Board Clock Generation

The 7RS110 includes on board oscillator and delay line to provide the high speed clocks to the R3000 components. The R3000 output signal SYSOUT is buffered and four copies are brought off the board for use as a system clock.

Cache Memory

Cache memory is provided on the module for 32 Kbytes of instruction cache and either 32K bytes or 16K bytes for data cache. Memory operations which require main memory data transfers are conveniently handled by means of on-board control signals.

Cache read miss operations are handled as single-word fetch and cache update. Non-cache read operations (such as I/O reads) are indicated by control signals.

Address and Data Buffers

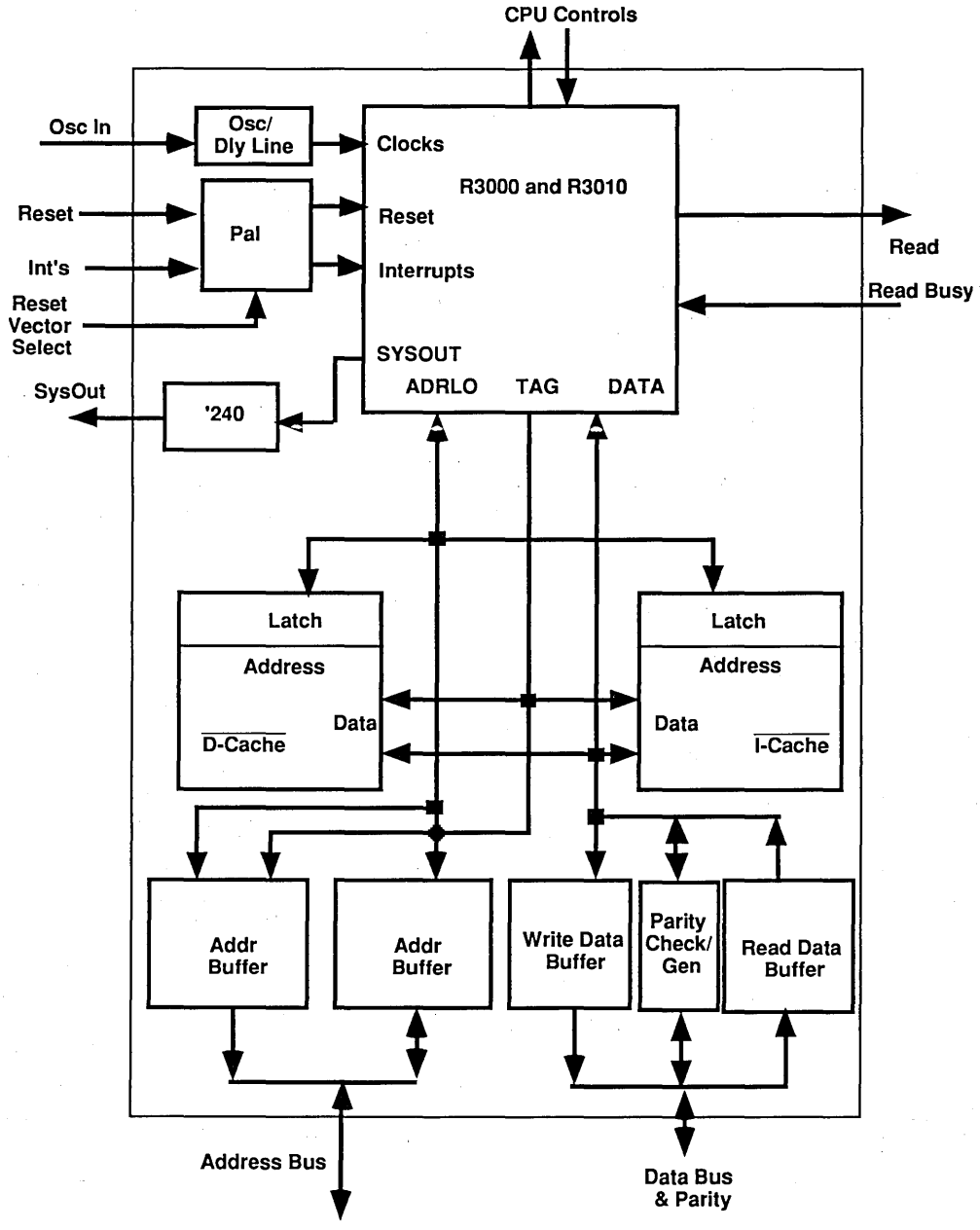
The address and data lines coming out of the module are buffered and can support substantial drive requirements. The Address Buffer is a bidirectional register, permitting cache snooping for multiprocessor systems.

The data pins are also driven by bidirectional registers. Memory write cycles utilize a single-word write buffer on the module which permits the R3000 Processor to continue running while data is being written into main memory.

R3000 User Interrupts

Six user interrupt inputs are provided, INT0-INT5. Each of these is a negative-true signal, terminated with a 10K ohm pullup load resistor on the module. The pins may be left unconnected if not used. The interrupt signals are synchronized in a PAL before being applied to the interrupt pins of the R3000 Processor.

FUNCTIONAL BLOCK DIAGRAM



SIGNALS PROVIDED ON MODULE PINS

Signal Name	Type	Description
MD00-MD31	I/O	Registered memory data lines to/from main memory system.
MDP0-MDP3	I/O	Parity bits for data lines.
MA00-MA31	I/O	Registered address output from the R3000 when used in a single-processor system. When used in a multi-processing system the bus is bi-directional.
MACT0-MACT1,	OUT	Positive-true outputs indicating the states of the R3000 outputs,
ACCTYP2		ACCTYP(0:2), which identify the nature of the data transactions for read/write cycles. MACT(0:1) are the registered ACCTYP(0:1) lines.
UINT0#-UINT5#	IN	Interrupt inputs.
EXTCLK	IN	External oscillator input (Needed when selecting external clock source).
CLKSEL	IN	Clock source selector. Should be pulled down to select external clock source.
MRES#	IN	Negative-true master reset input.
RES#	OUT	Negative-true reset output, connects directly to R3000 RES# pin.
SYSOUTA-D	OUT	Buffered clock outputs for synchronizing external events. Each of these is an identical clock signal, representing the inverted form of the R3000 output, SYSOUT#.
MEMRD#	OUT	Direct negative-true output from R3000, used to indicate that a memory read cycle is in progress.
MEMWR#	OUT	Direct negative-true output from R3000, used to indicate that a memory write cycle is in progress.
RBSY	IN	Positive-true input used to request a memory read stall initiation and termination. This signal is normally held in its asserted state and deasserted at the completion of stalls.
WBSY#	IN	Negative-true input used to request a busy indication for subsequent memory write operations.
CPC0	IN	Direct input to the R3000, used to indicate the size of the data (block, word, byte, or other) for memory read cycles.
CPC1	OUT	Connection between the R3000 Processcr and the R3010 FPA, indicating the status of the conditional branch. This pin is provided for diagnostic purposes, only.
CPC2, CPC3	I/O	Direct connections to R3000 pins.
EXC#	OUT	Direct output from R3000, indicating the EXC# signal between the R3000 and the R3010.
BERR#	IN	Negative-true input to R3000, used to indicate a bus error in main memory.
FPA#	OUT	Negative-true output for R3010, indicating the presence of R3010 FPA on the module.
FPINT#	OUT	Negative-true R3010 interrupt request. Connected on-board to INT1# or INT3# (jumper selectable).
RCOND(0:3)	IN	Reset condition lines. Used to select a reset vectors set from a given choices.
WPERR	OUT	Positive-true output used to indicate data parity error while write to memory operation.
RPERR	OUT	Positive-true output used to indicate data parity error while read from memory operation.
RDEN#	IN	Negative-true input used to latch data from the main memory bus in the read/write buffers.
WRCTL#	IN	Negative-true input used to latch data from the R3000 CPU in the read/write buffers.
WOE#	IN	Negative-true input used to write data from the read/write buffers to the main memory.
AEN#	IN	Negative-true input used to latch the R3000 address in the address buffers.
AOE#	IN	Negative-true input used to enable the output of the R3000 address from the address buffers to the main memory bus.
EXAEN#	IN	Negative-true input used to latch an external address in the address buffers.
EXAOE#	IN	Negative-true input used to enable the output of the external address from the address buffers to the R3000/cache bus.

RELATED PRODUCTS

Prototyping System

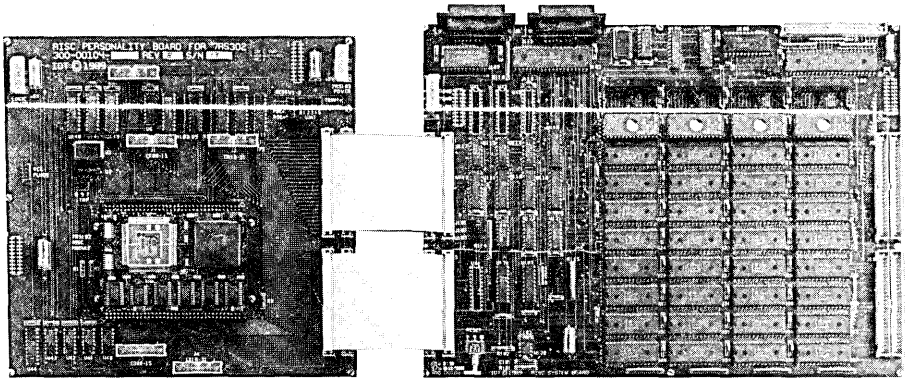
The 7RS110 module can be placed into immediate service using our flexible 7RS310 Prototyping Platform. The system includes two boards: a general purpose CPU board, and a personality card that interfaces the module to the CPU board.

The CPU board contains 1Mb of main memory, 256K of EPROM, two RS232 serial ports, an 8254 counter/timer, and an 8-bit parallel port accessible through a dual port RAM. Four 50-pin connectors provide access to all the address, data, and control signals for external connection to additional hardware on, for example, a wire-wrap board.

The system includes IDT's IDT/sim System Integration Manager, which provides facilities for downloading code, examining memory, and stepping through programs.

The personality card is on a separate board, and provides a bed for the module, necessary control signals, and connectors for an HP 16500 Logic Analyzer.

Code for the R3000 can be created on a MIPS development system, on IDT's MacStation™ system, or using IDT's Multi-Host cross assembler and compiler products, available for a variety of machines. Assembled code can be downloaded into the Prototyping System for execution and debug.



Module Prototyping Platform.
The card on the left is the personality card with a module; the card on the right is the general purpose CPU.

7

ORDERING INFORMATION

Ordering Part Number	CPU	FPA	I-cache	D-cache	Speed	Other
7RS110N55A16A	R3000A	R3010/A	32K	32K	16 MHz	
7RS110N55A20A	R3000A	R3010/A	32K	32K	20 MHz	
7RS110N55A25A	R3000A	R3010/A	32K	32K	25 MHz	
7RS110N55A33A	R3000A	R3010/A	32K	32K	33 MHz	
7RS110F55A16A	R3000A	R3010/A	32K	32K	16 MHz	
7RS110F55A20A	R3000A	R3010/A	32K	32K	20 MHz	
7RS110F55A25A	R3000A	R3010/A	32K	32K	25 MHz	
7RS110F55A33A	R3000A	R3010/A	32K	32K	33 MHz	
7RS110N54A16A	R3001	R3010/A	32K	16K	16 MHz	
7RS110N54A20A	R3001	R3010/A	32K	16K	20 MHz	
7RS110N54A25A	R3001	R3010/A	32K	16K	25 MHz	
7RS110N54A33A	R3001	R3010/A	32K	16K	33 MHz	
7RS110F54A16A	R3001	R3010/A	32K	16K	16 MHz	
7RS110F54A20A	R3001	R3010/A	32K	16K	20 MHz	
7RS110F54A25A	R3001	R3010/A	32K	16K	25 MHz	

ADDITIONAL INFORMATION

For more details on the 7RS110 module, refer to the 7RS110 Technical Specification and User's Manual.

RISC DEVELOPMENT SUPPORT

The R3000 RISC processors are blessed with the most powerful software development tools available for any machine. The MIPS workstation and server products make it possible to have many users on a well-controlled UNIX software development, executing and debugging code on native systems. IDT, as a MIPS partner, is an authorized reseller of these systems and the software support products that go with them.

Additionally, IDT has developed its own hardware and software development products which are more closely tuned with the needs of the small system or embedded control system designer. These include a low-cost, single-user R3000 workstation built around a Macintosh computer (the "MacStation"), and a variety of hardware prototyping tools.

For compiling C language programs, there is no better compiler anywhere than the MIPS C-compiler. The compiler and the CPU architecture were designed together to produce the highest performance possible running compiled C programs. Other languages, such as ADA and FORTRAN, are also supported on the MIPS systems. IDT's MacStation can be used to run the MIPS C-compiler as well.

IDT has independently developed a multi-platform cross C-compiler for the R3000 architecture. It is currently hosted on seven different platforms, including the ubiquitous 286 PC running MS-DOS. Although slightly less efficient than the MIPS C-compiler, it offers the advantages of readily available, inexpensive hosts and an integrated floating point library.

As with any other widely used processor, many support products have been developed by independent companies. These include ICE systems, Logic Analyzers and more. A partial list of third party suppliers is included at the end of this section.

TABLE OF CONTENTS

Integrated Device Technology, Inc.

IDT RISC DEVELOPMENT HOST SYSTEMS

PRODUCT OVERVIEW RC32xx

FEATURES:

- Based on Very High Performance MIPS® RISC Technology
- Binary compatibility across the complete line of MIPS RISComputers™, RISCstations™ and the IDT MacStation™ family
- Native Development Environment for MIPS ISA CPUs
- RISC/os™ MIPS SVID compliant UNIX® operating system with System V and BSD converged
- World class optimizing compilers including C, Pascal, ADA, PL/1, FORTRAN, COBOL
- Networking standards such as TCP/IP and NFS™ for connections to SUN, DEC, and other equipment
- X Terminal support via RISCwindows™
- Source level symbolic debugging
- Supports a wide range of development tools
  - — System Programmer's Package
  - — IDT floating point library
  - — IDT System Integration Manager
  - — Cache-2000 and Cache-3051 TargetSystem™ modeling tools for performance projections
- MIPS host platforms include:
  - — RC 3230/Magnum 3000 desktop workstation family
  - — RC 3240 deskside server family
  - — RC 3260 pedestal departmental server

DESCRIPTION:

This data sheet contains an overview of some of the tools and support available when developing microprocessor based systems using the MIPS development host platforms. These systems provide an easy-to-use, robust environment for the hardware and software development of today's high-performance RISC based systems, and greatly reduce the time and effort required to bring an application to market.

As in any development environment, ongoing enhancements are made to enrich the tool set. Detailed information on current products and enhancements, new tools, and third-party support products is available from your local IDT sales representative.

MIPS DEVELOPMENT SYSTEMS CHARACTERISTICS

MIPS' Development Systems, available from IDT, provide a large tool box when developing software and designing hardware architectures for any of the MIPS ISA CPUs. Based on the UNIX® operating system, RISC/os™ is SVID compliant, converging BSD and System V. Because RISC/os™ is a multiuser, multi-tasking operating system, many users can be active on the host at the same time and each user may initiate multiple tasks. These include simple multi-tasking capabilities from print spooling —the ability to queue listings to print while still compiling or editing— to more than 200 users all compiling, editing, linking, etc.

User protection indigenous to UNIX®, easily managed by the System Administrator, is a huge asset in developing, testing and maintaining systems software. Configuration Management, the job of software management, is much easier with these user isolation capabilities. For example, each user can belong to a group which is developing a certain section of the project. As a user builds and tests a subprogram, he submits that tested unit to Configuration Management for archive. The librarian can give other users access, but they cannot modify the code. This enforces coding rules as code is reviewed before it is accepted. The style may seem restrictive, but has many benefits: it ensures reliable code in the library, enforces testing procedures, and keeps code stable.

Many features of standard UNIX® operating systems ease the burden of program development. VI, the screen editor, (instead of a line editor), is a simple and powerful text editor that can be used for program entry. It has all the standard text editor features, plus an array of advanced capabilities for the experienced programmer. Saving and backing up important files is easily achieved by use of the TAR command. TAR, used for streaming tape backup, lists directories on tapes, can extract files from tapes, compare them, and manipulate them easily. Lastly, the Makefile command helps compile and link many subprograms easily. It is a simple list of all the files you need to compile and then link together. If during the debug process one subprogram is modified, invoking Makefile will find the single subprogram that has to be compiled again (by time/date stamps) and then relink automatically.

The CPU bandwidth required to be efficient at doing multitasking is well beyond the capability of today's medium range PC or Macintosh™ computers. Many advances in personal computer technology have stretched these single user architectures into multiuser domains, but to reduce the latency when adapting these systems requires more raw MIPS. These MIPS development hosts are thus ideal for the medium to large scale development team required for today's high-performance RISC-based applications.

CONNECTIVITY

The MIPS Development Platforms have been designed to easily integrate into existing computer environments. In most cases, these platforms easily integrate and allow the users to

8

access the system through their existing terminals, createsource files with existing terminals for download, compile and link, or more extensively use the systems file and printer resources. Solutions to a wide variety of integration goals and computing environments are readily available as parts of the MIPS Development platforms.

RS-232 Connections

The simplest method to interface to the MIPS Development Platform is to attach RS232 terminals directly. There are typically a number of direct connections, including Comm 0 for the system console (where the system is booted and system administration functions are often performed). These terminal ports allow a number of different RS-232 devices to be connected to the host system, including IBM compatible PCs running terminal emulation software or modem connections for telephone dialup. BAUD rates, parity, and other communication protocol information are easily configured.

Network Connections and X-Terminals

MIPS development systems are capable of integrating into all common computer networks, including Ethernet and DECnet. If the current environment includes thick or thin wire Ethernet the connection is simple, since MIPS systems support NFS-Network File System. NFS™ handles file transfers across a network, making all of the file resources of the network appear as if they belong in each system on the network. Since TCP/IP runs on the system, Ethernet file transfer and file sharing is transparent. This allows MIPS' systems to easily connect to DECstations, Sun workstations, and any other Ethernet system using these conventions. Using Ethernet, X Terminal support is provided by RISCwindows™. This is a MIPS native implementation of the standard X Window System™ combined with OSF Motif™. Alternately, the RISComm software package can be used to integrate the MIPS host into the DECnet computer network.

PC Compatible Connections

IBM PCs or compatibles can be connected directly through RS232 or Ethernet. For RS232 connections, a simple terminal emulator like Procomm Plus works well. The programmer can create source code using DOS applications and then transfer it to the MIPS platform using the Kermit protocol inside of Procomm Plus. Alternatively, the programmer can share files in the PC and the system by using products from Novell. They provide a software tool that runs in both systems allowing the user to compile files that exist on the PC, or run application programs on the PC and use the file system on the MIPS' platform transparently.

Macintosh

Several solutions are available to utilize existing Macintosh equipment. uShare™ provides a communication network between the Apple Macintosh and MIPS platforms. Once in place, it provides Mac users with server capabilities, electronic mail, print spooling, data backup, and terminal emulation allowing interactive work. This environment allows programmers to create code on the Mac using standard word processing applications, then transfer the file to the MIPS platform for compile and test. Caymen makes The Gator Box, which provides a gateway or bridge from Apple-Talk to the MIPS platform. Disk sharing and file sharing are capable with this tool as well.

IBM Mainframes

SNA 3270 allows RISC/os™ users to connect to IBM and others using SNA® protocol. Using the SNA 3270 software, the MIPS platform looks like an IBM terminal using any standard UNIX® terminal, eliminating the need for actual IBM terminals and printers. Support for standard IBM file transfer protocol for downloading and print spooling is possible with the session manager.

MIPS OPTIMIZING COMPILERS

A very important aspect of performance and development ease is the compiler technology. Fortunately, this is a fundamental strength of MIPS.

The MIPS instruction set and compiler technology were co-developed before any silicon was architected. Man years were expended developing, tuning, and testing the instruction set and the ability of compiler technology to generate efficient object code from high-level languages. After this rigorous analysis, the silicon tradeoffs were made. The result is a chip and compiler toolset designed to work with each other; this is dramatically different from traditional CPU/compiler development, where a chip manufacturer designs a chip and then expects compiler writers to "do their best with it".

MIPS optimizing compiler technology, widely regarded as the best in the industry, is based on mathematical models that work 100% of the time. This is substantially better than the "portable compiler" technology used for many other processors, which frequently uses heuristic rules for optimization that may not result in as robust operation. Today, five levels of compiler optimization are available. Each individual level further optimizes code size or execution speed. Local optimizations include: Optimal Calling Sequences, Branch to Branch Optimizations, Local Common Subexpression, Schedule FP Units, Pipeline Scheduling, Constant Folding, and Code Selection. Global optimizations, which optimize across procedure boundaries include: Moving invariant code out of loops, Strength Reduction, Register Assignment, Global Common Subexpression, Shrinkwrapping, Inter-procedural Register Allocation, and Procedure Inlining.

The structure of the MIPS compilers exemplifies "State of the Art" design; the compiler suite is architected to use a common optimizing structure, called the back end, and multiple language parsers or front ends. This foundation yields many benefits: improvement in the back end improves all languages, and more front ends (languages) can be easily added at any time and obtain the full benefit of MIPS optimizations. Also, the structure allows combinations of languages, including assembly, to be incorporated in a single program. MIPS compiler suite includes C, FORTRAN, Pascal, ADA, COBOL, PL/1, assembly, and plans to add C++ and ANSI C soon. Languages from third party vendors are available as well.

TOOLS FOR DEVELOPMENT

MIPS development hosts include support for both hardware planning, software development, and software integration. These tools allow application architects to define a system capable of meeting the cost/performance goals of the application, allow the software to be developed quickly and efficiently, and ease the process of integration of the application software onto the final target system.

Hardware Architecture Evaluation

The quality of an application system is determined by its ability to meet the needs of its marketplace. Good systems are well planned, well thought out, well architected, and are maintainable. Tools for hardware architecture definition help measure how close the final system will actually come to meeting the systems' predefined goal, before time and money are spent on developing prototypes. MIPS tools include the ability to accurately guage the performance of application software on a given memory architecture, and allow the hardware architect to make cost performance tradeoffs in the target system even before schematics are fully developed. These tools are contained in a product called the System Programmer's Package-SPP, and complement many of the software development tools discussed below. For example, Sable, a symbolic debugger/instruction simulator can be used to profile and develop kernel code. Pixie, Pixstats, and Profiler work to provide detailed information about the dynamic behavior of the software in the proposed system including information about cache performance, "hot spots" and dead spots in the code, and the dynamic instruction mix used by that software.

DBX Source-Level Debugger

This standard debugger for UNIX® is excellent for testing and finding software problems at the source code level. The debugger works for C, FORTRAN, Pascal, and assembly language. DBX is an easy tool to use; simply compile your code with the -g option (this keeps the symbol table from being stripped) and then invoke the debugger by typing "DBX command_file_name source_file_name". The command_file_name allows a set of DBX commands to run for set up, like setting variables, setting breakpoints, running to a specific point and displaying variables. Simple commands for DBX are: RUN, LIST, DUMP (list data about current procedure), STEP, STOP AT line #, STOP IN procedure_name, UP/DOWN (traverse stack activation levels), etc. Complex commands like "stop VAR in PROCEDURE if EXPRESSION" are also supported. DBX lets you use all the resources within the language except coprocessor zero, which can be debugged using "Sable".

Both the SPP and the IDT System Integration Manager allow the capabilities of DBX to be brought to source level debugging on a remote target system. This provides a familiar environment to the programmer as the task of integrating application software onto the target system is performed.

Program Behavior Analysis Tools

Pixie is a UNIX® program that provides statistics about a program's dynamic behavior. One type of data collected shows all the called routines, how many times they were called, how many instructions per call, and the percent of time spent in that call. This information is used to find out where most of the program time is spent, allowing the programmer to focus on those routines which dramatically affect the performance of the application. Another type of data collected shows the locality of the code. This helps to predetermine how much cache would be appropriate for this program. It also goes into detail on floating point usage, integer multiply/divide usage, all possible interlocks, load/branch nops, and more. For it to collect this type of data, it actually instruments your object code; the instrumented code is executed normally, and the output is formatted using the pixstats (pixie statistics) program.

The Profiler is another UNIX® tool that further defines code behavior. The user can concentrate in specific areas of program behavior by focusing the profilers attention to those items. Separate lists can be generated covering procedure information. They are: sorted by total time spent in each procedure, sorted by times called, sorted by number of cycles executed, sorted by number of clocks executed, sorted by number of clocks per line inside a procedure, and number of procedures never called. Again, this tool set allows the programmer to tune those areas of the code which will most affect the end system performance, thus increasing programmer efficiency.

The System Programmer's Package

SPP is a toolbox containing programs that help system developers to build, test, and download software to the target hardware. Among the utilities Included in the SPP are the following:

- Cache2000; a hardware environment simulator which allows the system architect to evaluate the performance of different types of memory systems
- Sable; a software environment simulator which allows symbolic kernel code debug
- DBGMON; a set of tools for downloading into the target machine, and interfacing to DBX to perform remote target source level debugging.
- common I/O routines and drivers used in many systems.

SPP is provided in source form and is written in C, allowing for easy modification and tailoring. SPP is covered in detail in its own data sheet.

SUMMARY

MIPS Development platforms provide all the tools necessary for hardware and software system design. UNIX complements the tool set by providing an engineering environment and communication capabilities to support a large or small engineering team. This environment supports the full development cycle required to be successful with and quick to market with applications based on the IDT/MIPS RISC processor family.

ADDITIONAL INFORMATION

Additional information about specific models and capabilities is available from your local IDT sales representative.

Integrated Device Technology, Inc.

PROTOTYPING PLATFORM FOR ANY IDT RISC CPU MODULE

IDT7RS300 Series

FEATURES:

- **Use with any module to form a complete R3000 based computer system**
- **Includes 1024K bytes of static RAM main memory, and 256K bytes of EPROM**
- **IDT's System Integration Manager in EPROM. Supports downloading software, single step, and other debugging needs**
- **Two RS232C serial ports (68681), parallel port (IDT7134 dual-port RAM), and programmable interval timer (8254)**
- **Address bus, data bus, and all necessary control signals provided on connectors (four 50-pin IDCs) to permit user expansion**
- **Direct connections to HP 16500 Logic Analyzer**
- **Fastest way to get an R3000 system up and running**

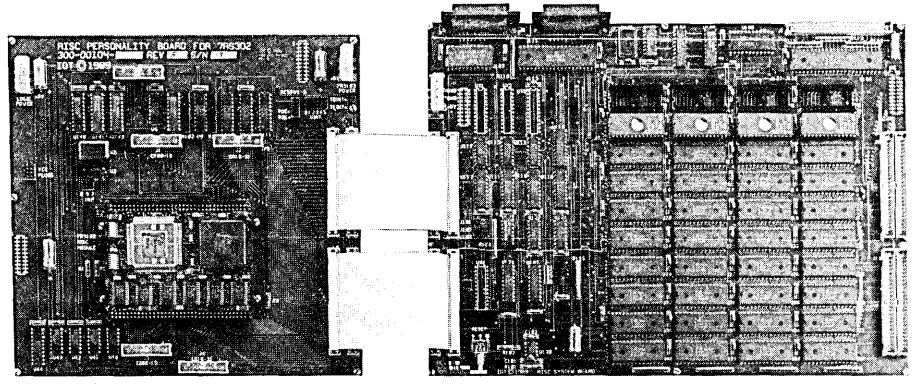
DESCRIPTION:

The IDT7RS300 series Prototyping Platforms consist of two separate PC boards which connect together by means of convenient ribbon cables to form the total system.

The System Board contains the non-CPU system functional units: static RAM main memory, EPROM monitor, two serial ports, a parallel port (utilizing an IDT7134 Dual-Port RAM), a free-running programmable timer, and expansion connectors to permit additional hardware to be added. It is designed to be placed on a flat table-top surface. Standoffs are provided for physical support. Access to all components on the board is readily available with this packaging approach. The Personality Board contains control logic and a socket for the a particular RISC module. The Personality Board is also designed for table-top use.

The EPROM on the System Board contains IDT's System Integration Manager (IDT/sim), a powerful tool for downloading software and debugging both hardware and software. Drivers can easily be added to support other I/O devices or changes in I/O addresses.

Software can be downloaded into the board from a MIPS machine, from an IDT MacStation development system, or from a PC running IDT's Cross Development Software.



The 302 version of the Prototyping Platform.
The System Board is on the right; the personality card and 102 module on the left.

FLEXIBLE PROTOTYPING PLATFORMS

System Description

The 7RS300 series RISC Prototyping Platform is designed to simplify the initial prototyping of both hardware and software for systems using one the the IDT RISC SubSystem Modules. The System Board is very general, and is the same in all of the 300 series Platforms. It contains basic control logic, mostly in PALs, 1 megabyte of static main memory, 256K of EPROM, a counter/timer, and I/O ports. Static RAM is used for main memory to provide the simplest interface to the module. The EPROM contains IDT's System Integration Manager in about 80K; the rest is available for user software.

The System Board connects to a personality board for the module through a pair of ribbon connectors. Each module architecture uses a different personality board. The personality board provides such features as clock generation, R3000 reset and initialization, read and write buffers, etc., to the extent that they are not already on the module. The personality board also contains five 20-pin plugs that can be directly connected to an HP 16500 series Logic Analyzer, and provides a uniform interface to the System Board.

System Board Hardware

The System Board is powered by a single 5 volt supply connected to a plug on the board. The plug conforms to the standard used for PCs, so an ordinary inexpensive PC power supply works easily with the board. A terminal can be connected to one of the RS-232 ports to act as the terminal for the Software Integration Manager. The other serial port is generally used to download software from some host system. Alternatively, there is an 8-bit wide parallel port built using dual port RAM that can be used for higher speed download.

Four 50-pin IDC (3M) connectors are configured for connecting additional hardware to the System Board. They contain the following signals:

- 32 bits of address
- 32 bits of data, and 4 parity bits
- SYSOUT (buffered clock from the R3000)
- RESET# (copy of the R3000's Reset signal)
- Parity and Address output enables from the address and data registers (to permit tri-stating other data onto these lines).
- Six interrupt lines to the R3000. These are registered or not, depending on the module.
- The four byte Write Enable signals.
- Five decoded chip select outputs from the upper 16 bits of address (1FE6 through 1FEE).
- MEMRD#, used to enable output devices in the expansion system during data read cycles.
- Auxillary input and output signals from the 68681 dual UART
- MREQ# and XACK# handshaking signals for controlling the timing of data transfers.

Personality Board Hardware

The personality board connects to the system board through two ribbon connectors. It contains a cut out area and plugs which accept the appropriate module. There are two five-volt power connectors, again using standard PC plugs. One power supply is for the personality card, the other for the module.

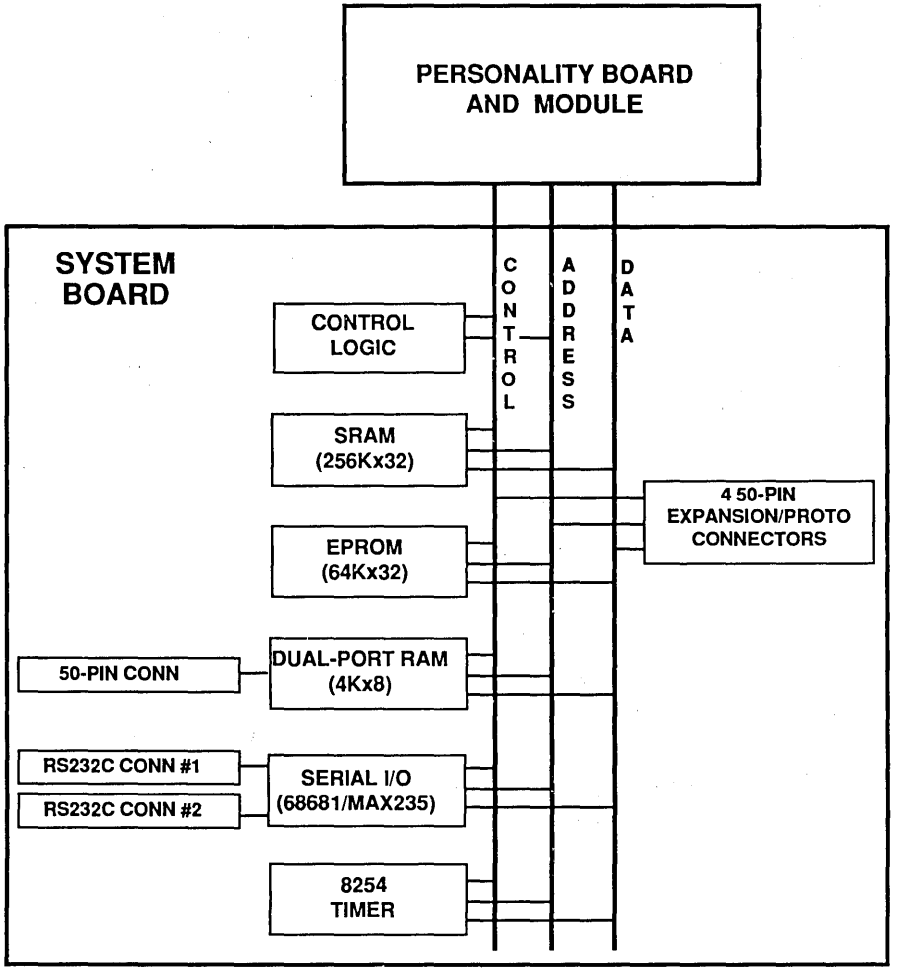
Five connectors are pre-wired to connect the modules signals to an HP logic analyzer. Because of the speed of the signals in the R3000 system, the connectors are placed on the slow side of the read/write buffers, so for disassembly and trace purposes, the R3000 must be run uncached.

Software Included

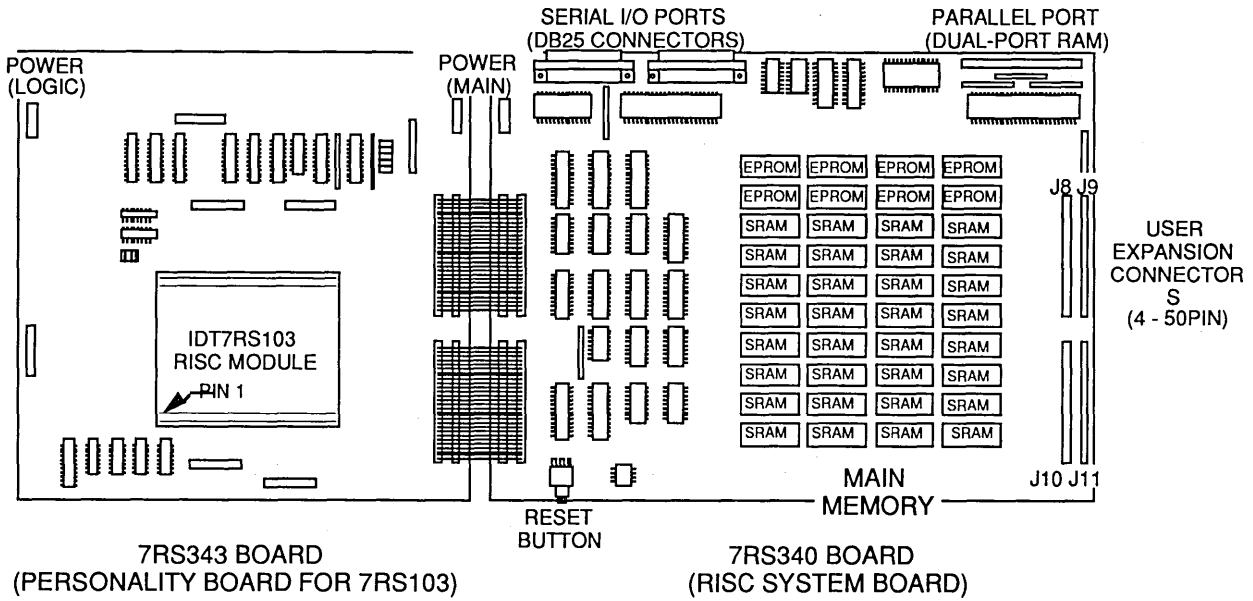
The System Board contains IDT's System Integration Manager (IDT/sim) in EPROM.

8

FUNCTIONAL BLOCK DIAGRAM



Block Diagram of the 7RS300 Series Prototyping Platforms



Layout of the series 300 Prototyping Platform.
The configuration shown is for the 7RS103 module.

ORDERING INFORMATION

Ordering Part Number	Speed
FOR USE WITH THE 7RS101 ARCHITECTURE	
7RS301-16	16 MHz
7RS301-20	20 MHz
7RS301-25	25 MHz
7RS301-30	30 MHz
FOR USE WITH THE 7RS102 ARCHITECTURE	
7RS302-16	16 MHz
7RS302-20	20 MHz
7RS302-25	25 MHz
FOR USE WITH THE 7RS103 ARCHITECTURE	
7RS303-16	16 MHz
7RS303-20	20 MHz
7RS303-25	25 MHz
FOR USE WITH THE 7RS104	
Contact Factory	
FOR USE WITH THE 7RS107 ARCHITECTURE	
7RS307-16	16 MHz
7RS307-20	20 MHz
7RS307-25	25 MHz
7RS307-33	33 MHz
FOR USE WITH THE 7RS108 ARCHITECTURE	
7RS308-25	25 MHz
FOR USE WITH THE 7RS109 ARCHITECTURE	
7RS309-25	25 MHz
7RS309-33	33 MHz
FOR USE WITH THE 7RS110 ARCHITECTURE	
7RS310-20	20 MHz
7RS310-25	25 MHz
7RS310-33	33 MHz

INCLUDED WITH SYSTEMS

Each Prototyping Platform includes the System Board, completely populated with 1 Mb of RAM and 256K of EPROM, with the Software Integration Manager in the EPROM. Each System also includes the appropriate personality card for the module architecture indicated and configured for the speed indicated. Documentation includes complete schematics for both the system board and the personality board, including all the PAL equations for the control circuitry.

Auxillary Download Program

For downloading code from a MIPS machine into an evaluation board. This software includes programs to convert MIPS object code into S-records and to download either ASCII or binary S-records to a remote target. This software is only needed with MIPS computers; all other machines (including the MacStation) have standard utilities available to perform this function.

MIPS download utility 7RS950BUU
Supplied on QIC-24 TAR tape.

Integrated Device Technology, Inc.

R3000 PGA ADAPTOR

IDT7RS363

FEATURES:

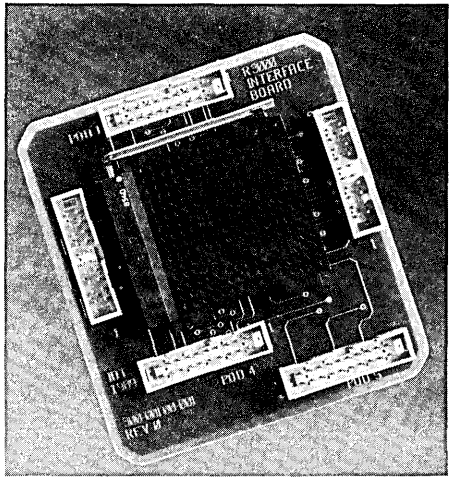
- **Simple and direct connection to HP 16500A Logic Analyzer System**
- **Probe points to 32 address, 32 data and 16 control signals of R3000**
- **Several clocks available for signal strobes**
- **Compact physical size permits its use in target system with minimal impact on spacing requirements**
- **Setup files for 16500A Logic Analyzer assures speedy startup**
- **No active components**
- **Compact design assures minimal added lead capacitances (approx 5 pF)**

DESCRIPTION:

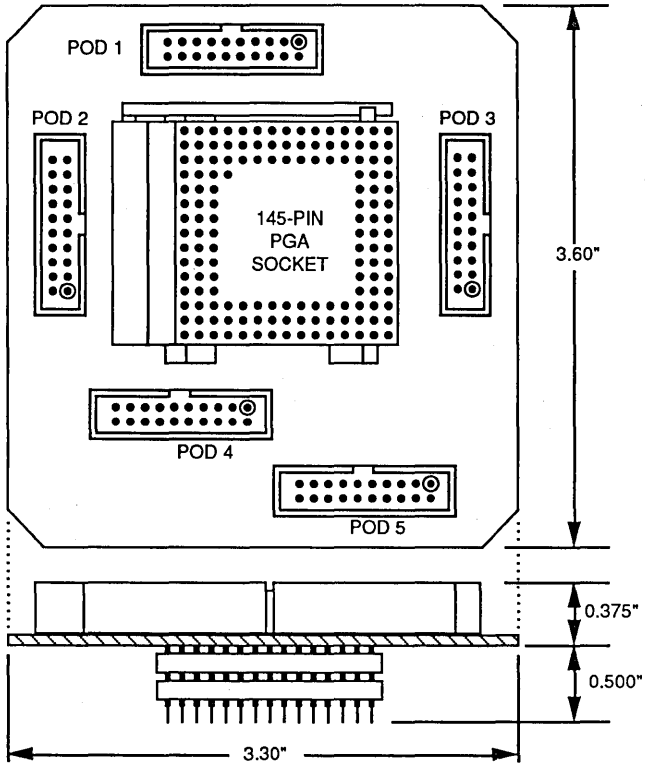
The IDT7RS363 is an adapter card intended for use in performing diagnostics on the operation of the IDT79R3000 RISC Processor on a Hewlett-Packard model 16500A Logic Analysis system. It contains no active components. Instead, it is used as a socket adapter for the R3000, and all address, all data, and many control lines are made accessible for capture by the logic analyzer. It may only be used with the pin grid array (PGA) package of the R3000 and requires the logic analyzer system to be equipped with 5 HP Termination Adapters, P/N 01650-63201, to provide for the direct connection to the analyzer input pods.

For ease of setup, a diskette is provided, as a part of the 7RS363, which contains files loadable directly into the HP logic analyzer. These files automatically set up the logic analyzer by assigning the pods and the individual input channels directly to the signals captured from the R3000. In this way, the logic analyzer display will immediately represent the captured signals with all the proper signal names displayed.

The 7RS363 may also be used as a simple diagnostic tool, separate from its use as a logic analyzer adapter card. This is accomplished by virtue of the fact that all necessary signals of the R3000 are immediately accessible as test points on the card.



8



ORDERING INFORMATION

The PGA Adaptor is shipped with documentation describing all the pin connections.

PGA Adaptor for R3000 ..... 7RS363

Integrated Device Technology, Inc.

R3000 DISASSEMBLER FOR USE WITH THE HP 16500 LOGIC ANALYZER

IDT7RS364

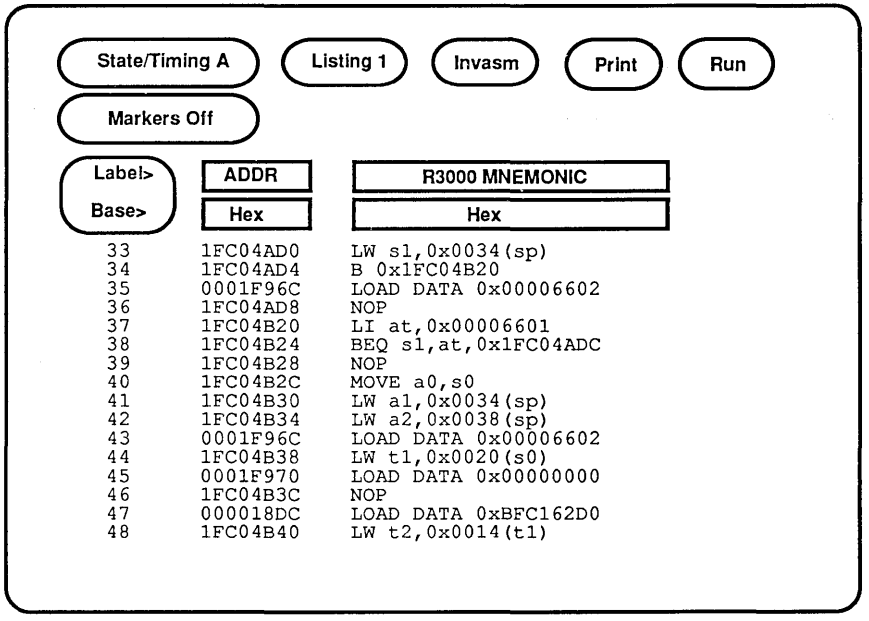
FEATURES:

- Disassembler for the IDT79R3000 RISC microprocessor on the HP16500 logic analyzer
- Interfaces to the IDT7RS301 And IDT7RS302 TargetSystem Boards
- Allows actual code tracing
- Direct Connection to IDT 7RS300 series Prototyping Platforms
- Can Be Used with Any R3000 System

DESCRIPTION:

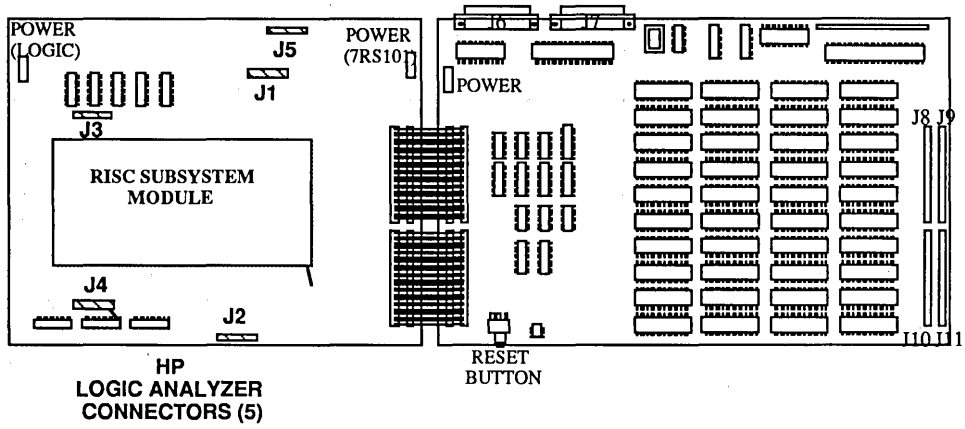
The IDT7RS364 Disassembler for the HP16500 logic analyzer is a useful tool meant to ease the task of debugging software run on the IDT7RS301/302 TargetSystem Boards. The HP16500 itself allows the engineer to capture the CPU's executed opcodes. However, the IDT7RS364 will allow the engineer to trace the actual code executed by the processor without having to resort to lookup tables. The IDT7RS364 will decode the opcodes captured by the logic analyzer into the correct instruction mnemonic so that the engineer will know the exact processor state.

The IDT7RS364 will only capture main memory accesses (see Figure 1). Therefore its operation is guaranteed only if the software is being run in uncached address space.



Typical Screen Display

8



The 7RS301 Prototyping Platform for the 7RS101 Modules.
The five connectors for the logic analyzer are on the module personality card.

ORDERING INFORMATION

The Disassembler is shipped on a 3.5" diskette for the HP16500 series Logic Analyzers. Included is complete set-up documentation and drawings showing how to use it with systems other than the 7RS300 series Prototyping Platforms.

Disassembler 7RS364

Integrated Device Technology, Inc.

R3000 AND R3001 EVALUATION BOARDS

IDT7RS382
IDT7RS383

FEATURES:

- **Complete RISC Example System**
- **Available for R3000 or R3001 CPUs**
- **Supplied with complete schematics and PAL equations**
- **Includes R3010 Floating Point Accelerator, Data and Instruction Caches (16k bytes each), 128Kb of main memory, and 128Kb of EPROM**
- **IDT's System Integration Manager included in EPROM**
- **Connectors provided for easy expansion**
- **CPU and FPA in PGA Sockets for easy connection to Logic Analyzers**

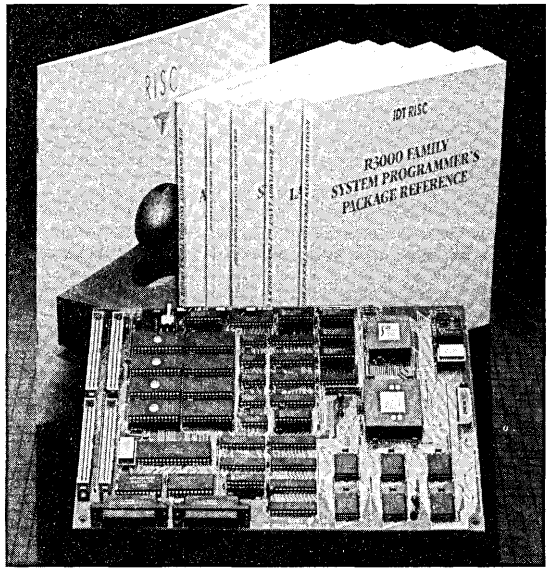
DESCRIPTION:

The IDT7RS382 and 7RS3833 are complete RISC systems intended for use as low-cost evaluation systems They are completely self-contained on a single printed circuit board and requires only a simple CRT terminal and a 5 volt power supply for operation.

The board contains the RISC CPU (R3000 on the 7RS382 and R3001 on the 7RS383) and a IDT79R3010 Floating Point Accelerator (FPA), as well as 16 Kbytes of Cache RAM for each of the data and instruction caches, EPROM, static main memory, two serial ports, a free-running programmable timer, and expansion connectors to permit external systems to be added. In addition, the system also contains clock timing, reset initialization logic, and read/write buffer control logic.

The EPROM contains IDT's powerful System Integration Manager (IDT/sim), a debugging monitor that supports download of code from host systems, execution control commands, memory probing, and I/O.

The board is 11 inches long and 7.5 inches wide and is designed to be placed on a flat table-top surface. Standoffs are provided for physical support. Access to all components on the board is readily available with this packaging approach.



8

COMPLETE SINGLE BOARD COMPUTERS

The 7RS382 and 7RS383 Evaluation Boards are complete single board RISC computers, requiring only a 5 volt power supply and an RS-232 terminal for operation.

Both systems are intended as examples of typical designs using the R3000 and R3001 CPU chips. All the schematics and details of the designs are supplied with the boards.

The boards can also serve as a prototyping platform. Connectors are provided to add additional hardware on a wire wrap board. A 20-pin plug, J5 on the drawings, provides 16 control signals that can be connected to a logic analyzer. Both the CPU and the FPA chips are mounted in PGA sockets, so a PGA adaptor (7RS363) can be used to connect all the CPU pins to a logic analyzer.

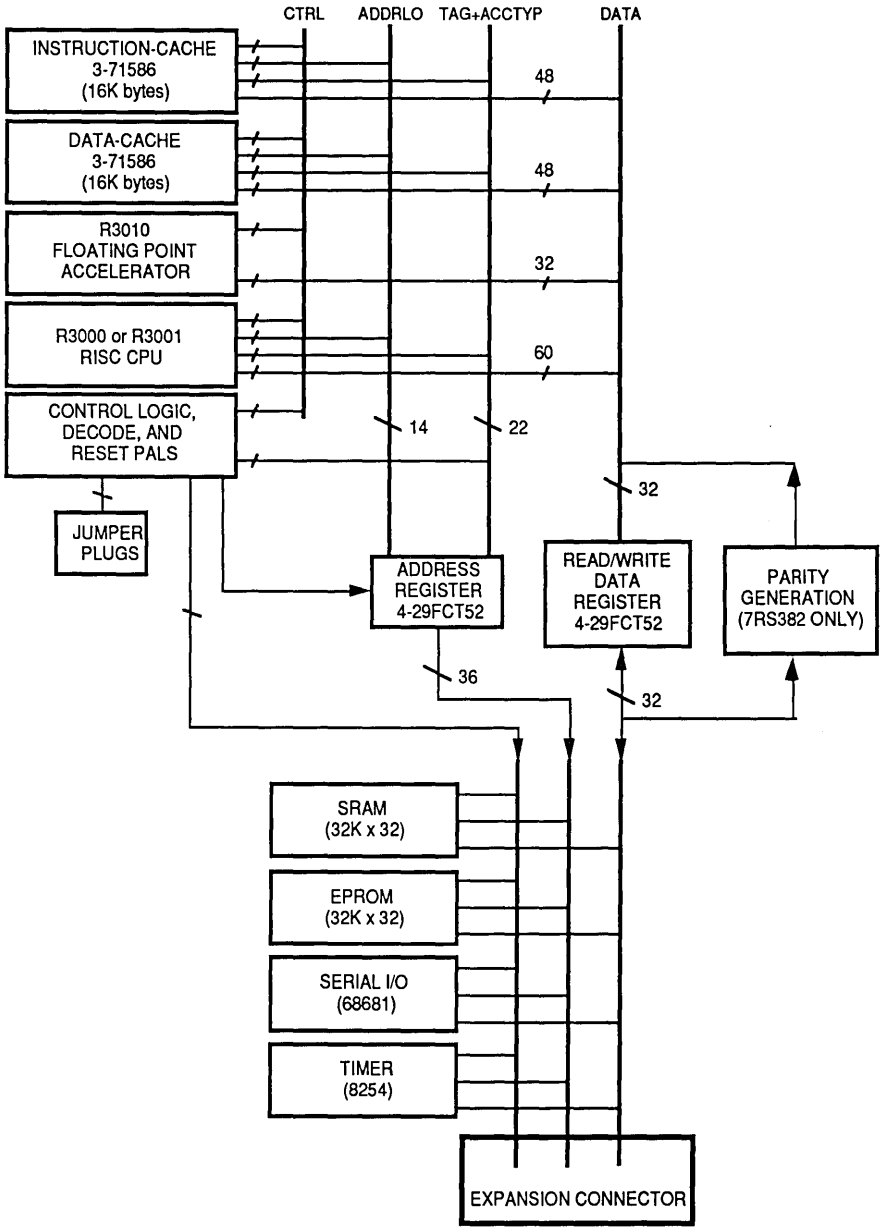
The 7RS383 (but not the 7RS382) can accept larger RAM and EPROM devices, supporting up to 512 Kb of EPROM and 2 Mb of static RAM.

Software supplied in the EPROM permits downloading of R3000 code compiled on a PC, MacStation, or MIPS machine. Commands are available to set break-points, single step, examine and modify memory, etc.

SIGNALS SUPPLIED ON EXPANSION CONNECTOR

Signal Name	I or O	Description
EA00-EA31	I/O	32-bit buffered address bus
ED00-ED31	I/O	32-bit buffered data bus
SYSOUT	O	Buffered SYSOUT Clock from CPU; used to synchronize data transfers
MRES#	O	Copy of the Reset signal to the CPU
MREQ	O	Memory Request output (handshaking signal for data transfers)
XACK#	I	Acknowledge input (handshaking for data transfers)
OP3-OP5	O	Auxillary output pins from the 68681 UART
IP4-IP5	I	Auxillary input pin to the 68681 UART
WEA-WED	O	Write Enables for the four bytes of the data word
UCS1A-UCS1D	O	Four chip select signals decoded from the high order address bits.
INT0	I	Interrupt input to the R3000
PMRD#	O	Memory Read output signal. Used to enable output drivers in the expansion system during data read operations

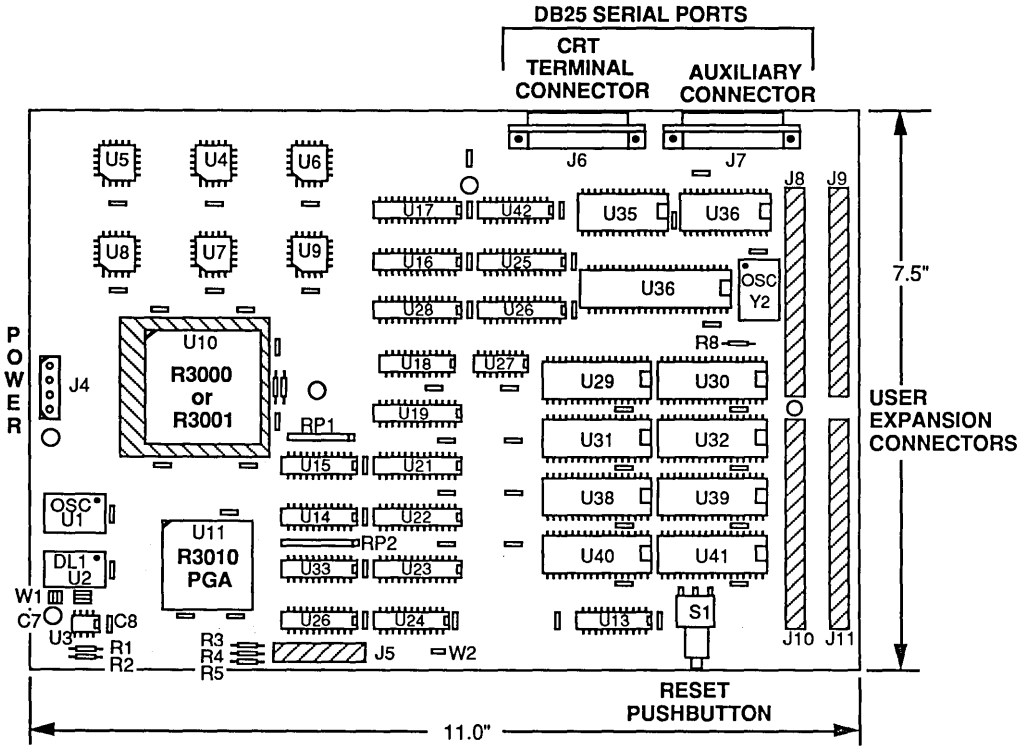
FUNCTIONAL BLOCK DIAGRAM



The 7RS383 (with R3001) does not require parity; the 7RS382 (with R3000) includes parity generation logic on the board.

8

PHYSICAL LAYOUT



(U numbers may vary)

ORDERING INFORMATION

Each unit is shipped with complete schematics and PAL equations. A user's manual includes instructions on downloading code, operating the Software Integration Manager, and providing the correct timing interfaces to additional hardware.

Evaluation Boards

R3000 Based Evaluation Board 7RS382
R3001 Based Evaluation Board 7RS383

EPROM Upgrades

The following part numbers are for updating the evaluation board hardware to the latest version of the monitor.

For Evaluation boards and Prototyping Systems 7RS901BAP
Use with 7RS382,383, or any 7RS300 series module Prototyping Platform

Auxillary Download Programs

For downloading code from a MIPS machine into an evaluation board. This software includes programs to convert MIPS object code into S-records and to download either ASCII or binary S-records to a remote target. This software is only needed with MIPS computers; all other machines (including the MacStation) have standard utilities available to perform this function.

MIPS download utility 7RS950BUU
Supplied on QIC-24 TAR tape.

8

Integrated Device Technology, Inc.

REAL8™ R3000 LASER PRINTER CONTROLLER EVALUATION SYSTEM

IDT7RS388

FEATURES:

- **Plugs Directly into Canon LBP-SX 8-ppm print engine**
- **Provides approximately 8x performance of LaserWriter II NTX on typical benchmarks**
- **25 MHz R3001 RISController™ CPU with R3010 Floating Point Accelerator and 16K each of Instruction and Data Cache**
- **Includes 4 MB of DRAM, socketed**
- **Supports up to 4 MB of EPROM, socketed**
- **Software Included on EPROM**

  TrueImage™ Page Description Language

  PeerlessPage™ Printer Operating System

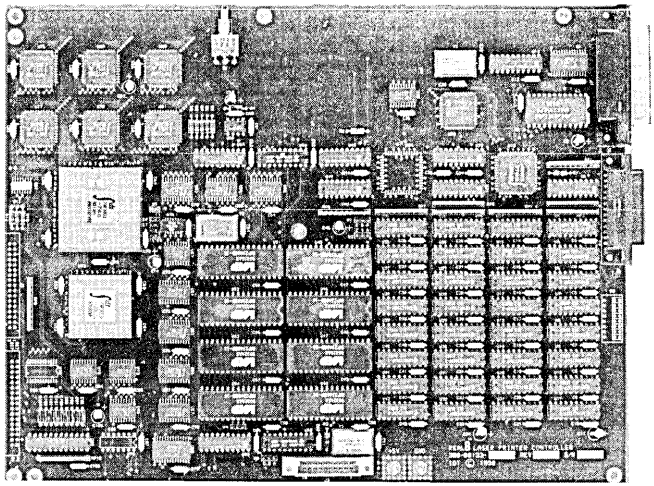
  IDT/sim Debug Monitor
- **Schematics and Hardware Reference Manual included**

DESCRIPTION:

IDT's 7RS388 REAL8 Laser Printer Controller provides a board level, hands-on, design demonstration and benchmarking tool for IDT R3000 based, PostScript™, TrueImage, or PCL5 class Raster Image Processing (RIP).

The REAL8 uses a 25 MHz R3001, with 16KB Instruction and 16KB Data Caches to rasterize pages at performance points from four to twelve times 68020 engine-based controllers. The board is supplied with TrueImage PDL and fonts for direct speed comparisons with other controllers. All the design documentation is supplied, and the design is easy to tailor to a variety of cost/performance levels by adding additional I/O or deleteing RAM or EPROM. A demonstation copy of PeerlessPage Printer Operating System is also included. PeerlessPage can support a variety of I/O drivers and Page Description Languages, including multiple PDLs in the same controller.

The REAL8 controller fits into the industry standard Canon LBP-SX laser print engine and provides standard LBP-SX "video" output to the engine.



The REAL8 Controller Board

PRODUCT OVERVIEW

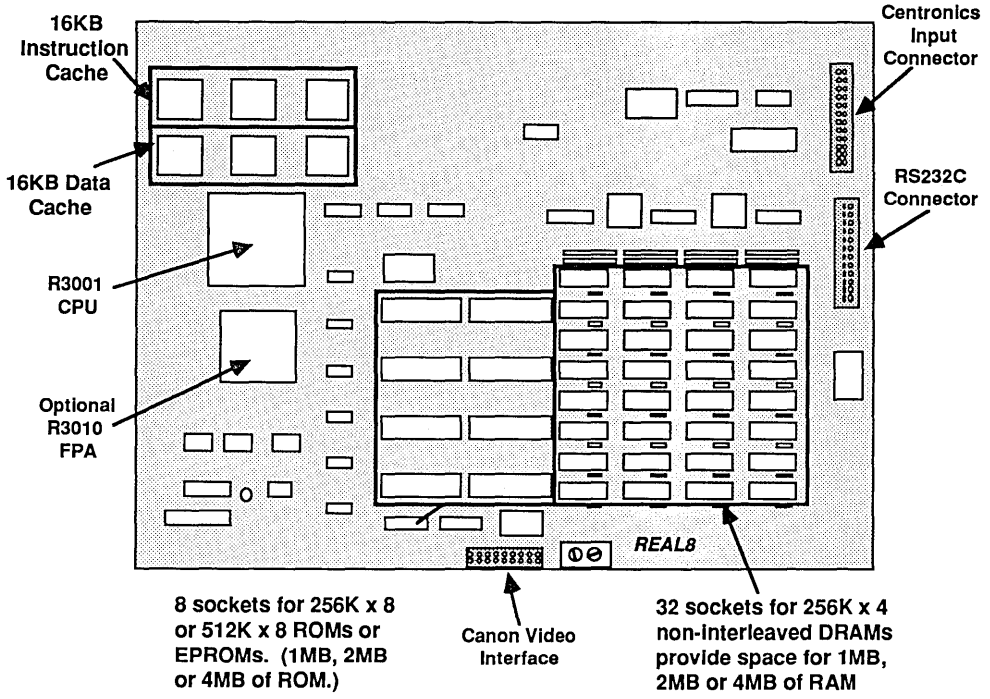
The IDT7RS388 REAL8 Printer Controller is an evaluation vehicle allowing designers to run benchmarks to compare the performance of an R3000 based controller with those built around other processors. The product includes complete design documentation, so it provides a design model which can be enhanced with other I/O ports, emulations, cartridge connectors, front panel controls or other differentiating features. The PeerlessPage Printer Operating System included with the board makes it easy to interface new I/O drivers and new printer emulations.

The REAL8 board runs at 25 MHz, but the same design could be used, with appropriate component selections, over a range of 16 to at least 33 MHz.

The ROM and DRAM memory areas provide sockets for user experimentation with configurations up to 4MB of each. There is adequate RAM space for up to two page buffers (at 300 dpi), and program/font space in ROM/EPROM for PostScript®, TrueImage, PCL5, combinations of emulations, or increased font storage. The SX video oscillator can also be changed to provide 400 dpi output.

Benchmarking files can be downloaded to the REAL8 from any IBM/PC or PC compatible over the standard Centronics parallel printer port by simply copying each file to LPT1; the original source application program does *not* have to be running.

REAL 8 LASER PRINTER CONTROLLER



IDT7RS388 Controller Fits Canon LBP-SX Print Engine

8

DEMO SOFTWARE INCLUDED

The REAL8 EPROMS contain a demo copy of Microsoft TrueImage PDL, TrueType fonts, and the PeerlessPage Printer Operating System from The Peerless Group. As shipped, the board can be plugged directly into a Canon SX engine and can print downloaded PostScript files. This provides an easy way to compare the performance of this R3000 based controller with controllers based on conventional technology or other RISC machines.

Microsoft TRUEIMAGE™ PDL

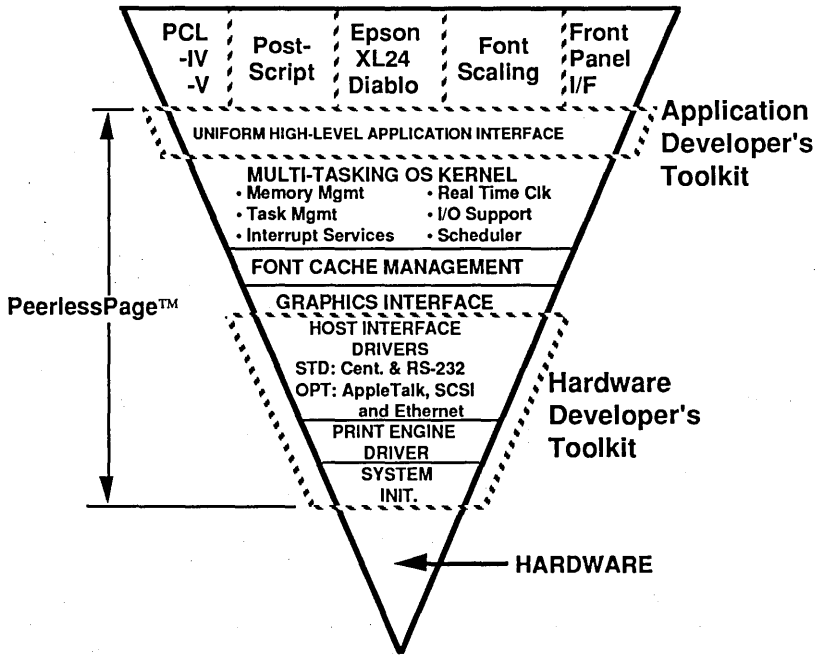
TrueImage is an open technology Page Description Language for high performance full-function printers. TrueImage provides extensive features such as TrueType™ scalable fonts, enhanced communication with the Microsoft Windows™ operating environment and improved printer performance, as well as complete Adobe® PostScript compatibility. TrueImage contains a TrueType rasterizer, as well as an Adobe Type I rasterizer, and is able to execute Type I fonts.

PEERLESSPAGE™ Printer OS

PeerlessPage provides a real-time multitasking executive/kernel surrounded by graphics and I/O services such as font and emulation soft-switching and active port sensing. PeerlessPage is a portable, easily extensible foundation that supports multiple industry standards and accelerates time-to-market by enabling OEMs to quickly and cost-effectively build differentiated product lines.

The PeerlessPage OS itself has been ported to the REAL8 board and is included in the board's EPROMs. The TrueImage PDL interpreter runs on top of PeerlessPage. Other PDLs can easily be ported to the REAL8 by taking advantage of the graphics and font handling features built into PeerlessPage.

The PeerlessPage Interface manuals are available directly from the Peerless Group, and describe the calls available to the user. Users can write C programs with the appropriate calls, download the compiled code to the REAL8 board, and execute it in a software development environment.

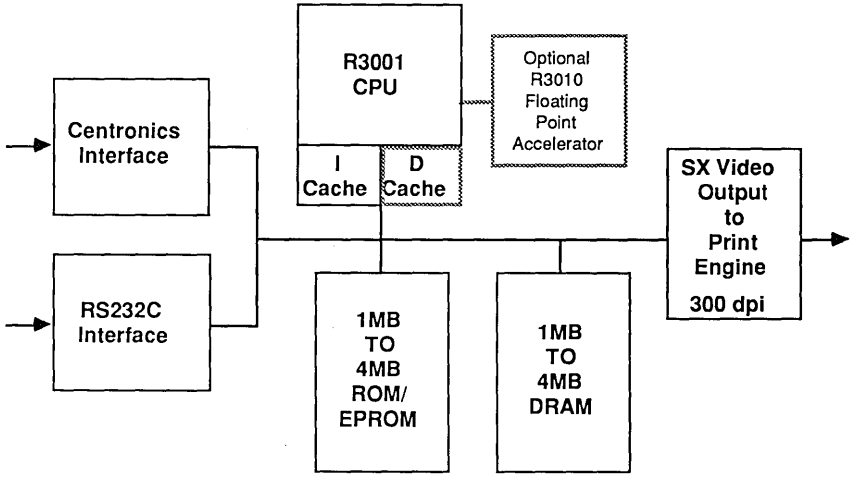


PeerlessPage Printer Operating System

ARCHITECTURAL HIGHLIGHTS

The IDT7RS388 uses a 25 MHz R3001 RISController™ with optional (socketed) R3010 Floating Point Accelerator. The FPA typically provides 15 - 20 % performance improvement in PostScript applications and is not required at all for PCL applications.

The IDT7RS388 assumes that Instruction Cache is always used, but provides a jumper for disabling the Data Cache. Similarly, junpers allow the ROM/EPROM area to accommodate different memory configurations (see Hardware Reference Manual included with Kit).



ORDERING INFORMATION

Each REAL8 Laser Printer Evaluation System includes a controller board for a Canon LBP-SX engine with 4 MB of DRAM and with PeerlessPage and TrueImage in EPROM. Also included is a hardware reference manual describing the design in detail, including schematics and PAL equations.

REAL8 Printer Evaluation Board 7RS388

8

Integrated Device Technology, Inc.

MacStation™ 2
R3000 DEVELOPMENT SYSTEM

IDT7RS502

FEATURES:

- 10 VAX MIPS RISC Computer inside any Macintosh II
- Includes UNIX V 3 Operating System
- Supplied with MIPS C-Compiler, Assembler, and Symbolic Debugger
- Will support any MIPS software packages, including SPP, SPP/e, FORTRAN
- Uses all Macintosh peripherals for I/O
- Multifinder and System 7 Compatible

R3000 COMPUTER PLUGS INTO A MACINTOSH II:

The MacStation 2 is an R3000 based workstation consisting of a Macintosh II computer and a high-performance R3000 CPU that plugs into the NuBus inside the Mac. The R3000 CPU runs IDT/ux, IDT's port of MIPS RISC/os UNIX V 3, in a window under Multifinder.

The UNIX window is opened by double-clicking an icon on the Mac screen. The window is essentially a terminal emulator. When the window is opened, IDT/ux starts up on the R3000 CPU. Users can switch back and forth between the IDT/ux window and other Macintosh windows freely.

The system includes MIPS powerful C Compiler. Source code is written using any Macintosh text editing program, and a single command in the IDT/ux window will copy the file into the UNIX file system and compile it. Any other software written for MIPS RISC/os can be ported to the MacStation 2.

The MacStation hardware uses two slots in the Macintosh. The MacStation software is supplied on tapes and requires approximately 160 Mb of hard disc space when loaded.

MacStation 2

MacStation 2 HARDWARE

RISC CPU Card

The CPU card is a 16 MHz R3000 system, using the Floating Point Accelerator and 64 Kb each of instruction and data caches. The necessary hardware to run UNIX is also on the board. EPROMs on the card contain IDT's System Integration Manager (IDT/sim) which provides many debug and control features at the monitor level. The board communicates with all I/O via the Macintosh NuBus.

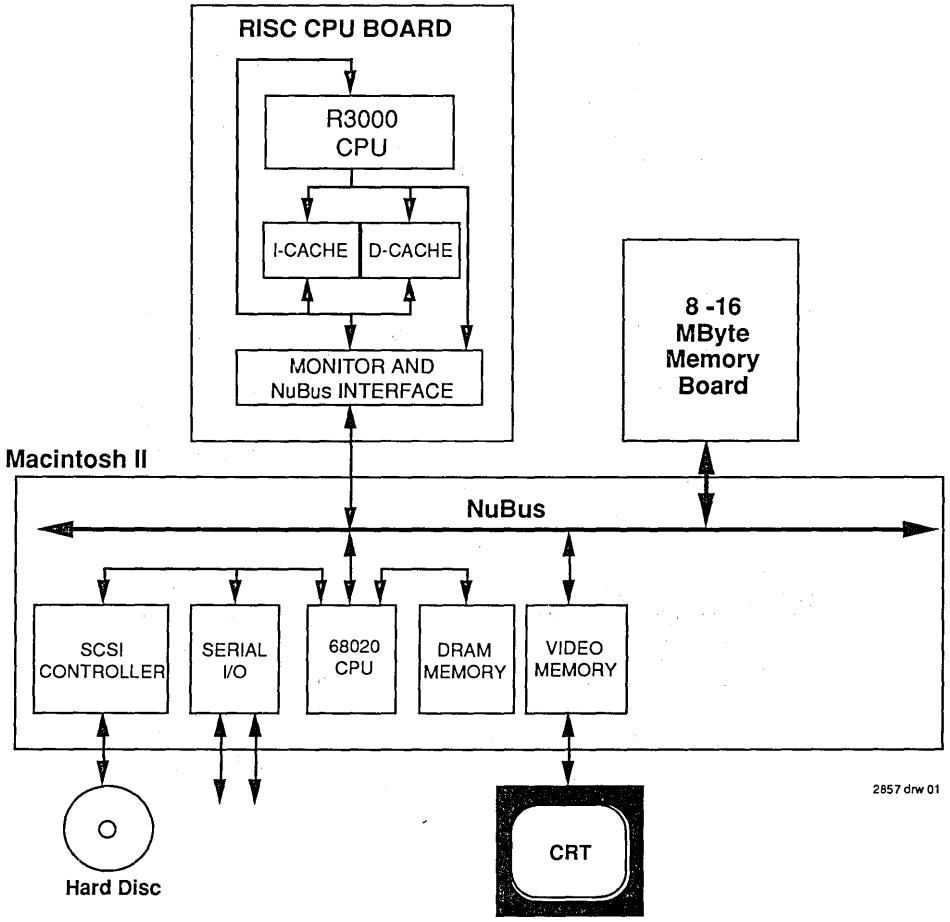
Main Memory

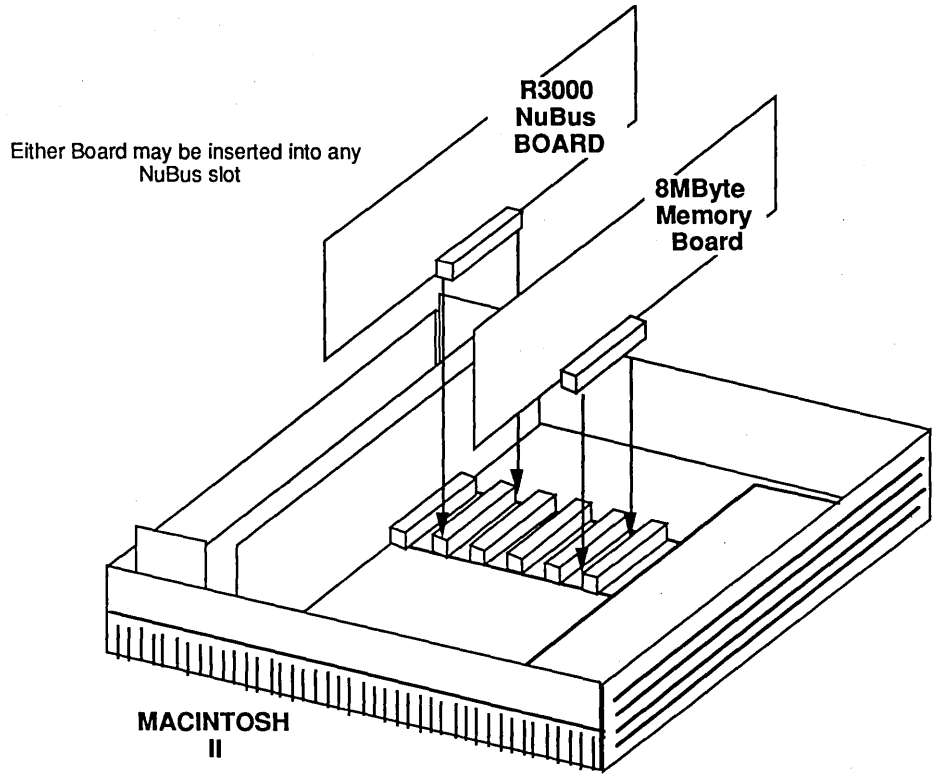
The MacStation 2 includes 8 Megabytes of memory on a separate NuBus card. This permits reasonably large programs to execute without excessive disc swapping. The memory may be expanded to 16 Mb by simply adding an 8 Mb extension card, available separately.

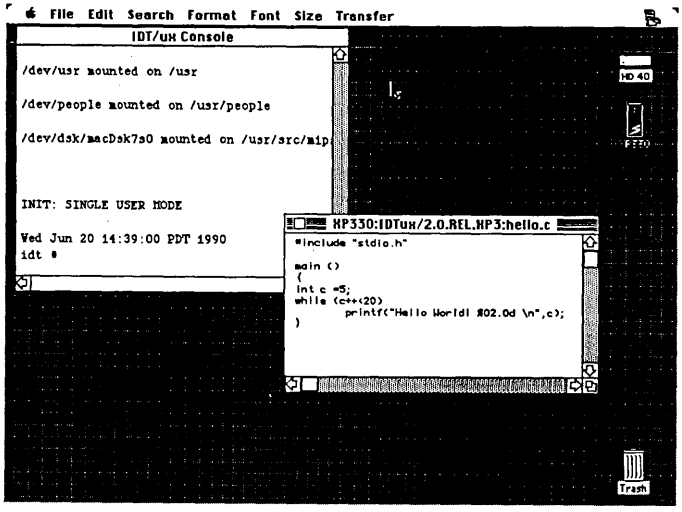
MacStation 2 SOFTWARE

The MacStation is shipped with all the software on tapes. It may be ordered with a 160 Mb hard disc with the software pre-installed, but tapes are still included for backup.

FUNCTIONAL BLOCK DIAGRAM







Screen Shot of Macintosh with IDT/ux running.

ORDERING INFORMATION

The MacStation 2 may be ordered in a variety of forms, ranging from the essential boards and software to complete systems with software pre-installed. The IDT/ux operating system requires a signed single-system license agreement which must be executed before the system can be shipped. Contact your IDT sales office for a sample of the license.

MacStation Boards 7RS502B8-L
Includes R3000 NuBus CPU card, 8 Mbyte NuBus Memory card, IDT/ux. Requires a Macintosh II computer, system 6.01 or later, Apple tape drive, at least 160 Mbytes of free disc space.

MacStation Conversion Kit 7RS502TD8-L
Everything to convert a Macintosh II to a MacStation. Includes the MacStation boards, IDT/ux, a tape drive, a 160 Mbyte Hard Disc and cables.

Memory Expansion Card 7RS502X8
Adds 8 Mbytes to the Memory Card, raising total available to IDT/ux to 16 Mbytes.

Standard MonoChrome System 7RS502MXM-L
Includes the MacStation Boards, IDT/ux, tape drive, 160 Mb external hard disc, cables. Also includes Macintosh IIx computer with 4 Mbytes of internal memory and 80 Mb internal hard disc, monochrome 13" monitor and video card, extended keyboard.

High Performance MonoChrome System 7RS502MFXM-L
Includes the MacStation Boards, IDT/ux, tape drive, 160 Mb external hard disc, cables. Also includes Macintosh IIfx computer with 4 Mbytes of internal memory and 80 Mb internal hard disc, monochrome 13" monitor and video card, extended keyboard.

Other Macintosh system configurations.
The MacStation can be supplied with any desired Macintosh configuration. Contact factory for a quotation.

IDT/ux Documentation

This is an eight-manual set of documentation for MIPS RISC/os (UNIX V 3 with BSD extensions). Included are: RISC/os Programmer's Reference Manual, User's Reference Manual, System Administration Reference Manual, System Administrator's Guide, Programmer's Guide, User's Guide, Streams Primer and Programmer's Guide, Guide to BSD on RISC/os.

IDT/ux Documentation Package 7RS551BDU

Other MIPS Software for MacStation

The following MIPS products are available for the MacStation. All require asigned license agreement prior to shipment. Contact your IDT sales office for a sample of the agreement.

SPP for MacStation 7RS992SMT-L
Source Code, site license, includes documentation

SPP/e for MacStation 7RS993SMT-L
Source Code, site license, includes documentation

8

Integrated Device Technology, Inc.

MacStation 3
R3000 DEVELOPMENT SYSTEM

ADVANCE INFORMATION
IDT7RS503

FEATURES:

- 20 VAX MIPS RISC Computer Inside any Macintosh II
- Includes UNIX V 3 Operating System
- Supplied with MIPS C-Compiler, Assembler, and Symbolic Debugger
- Will support most MIPS software packages, including SPP
- Uses all Macintosh peripherals for I/O
- Includes Macintosh-Independent SCSI and serial I/O ports
- Multifinder and System 7 compatible

R3000 COMPUTER PLUGS INTO A MACINTOSH II

The MacStation 3 is a high performance R3000 based workstation consisting of a Macintosh II computer and an R3000 CPU board that plugs into the NuBus inside the Mac. The R3000 CPU runs IDT/ux, IDT's port of MIPS RISC/os UNIX V 3, in a window under Multifinder.

The UNIX window is opened by double-clicking an icon on the Mac screen. The window is essentially a terminal emulator. When the window is opened, IDT/ux starts up on the R3000 CPU. Users can switch back and forth between the IDT/ux window and other Macintosh windows freely.

The system includes MIPS powerful C Compiler. Source code is written using any Macintosh text editing program, and a single command in the IDT/ux window will copy the file into the UNIX file system and compile it. Any other software written for MIPS RISC/os can be ported to the MacStation.

The MacStation hardware uses one NuBus slot in the Macintosh. The MacStation software is supplied on tapes and requires approximately 160 Mb of hard disc space when loaded.

MacStation 3

MacStation 3 HARDWARE

RISC CPU Card

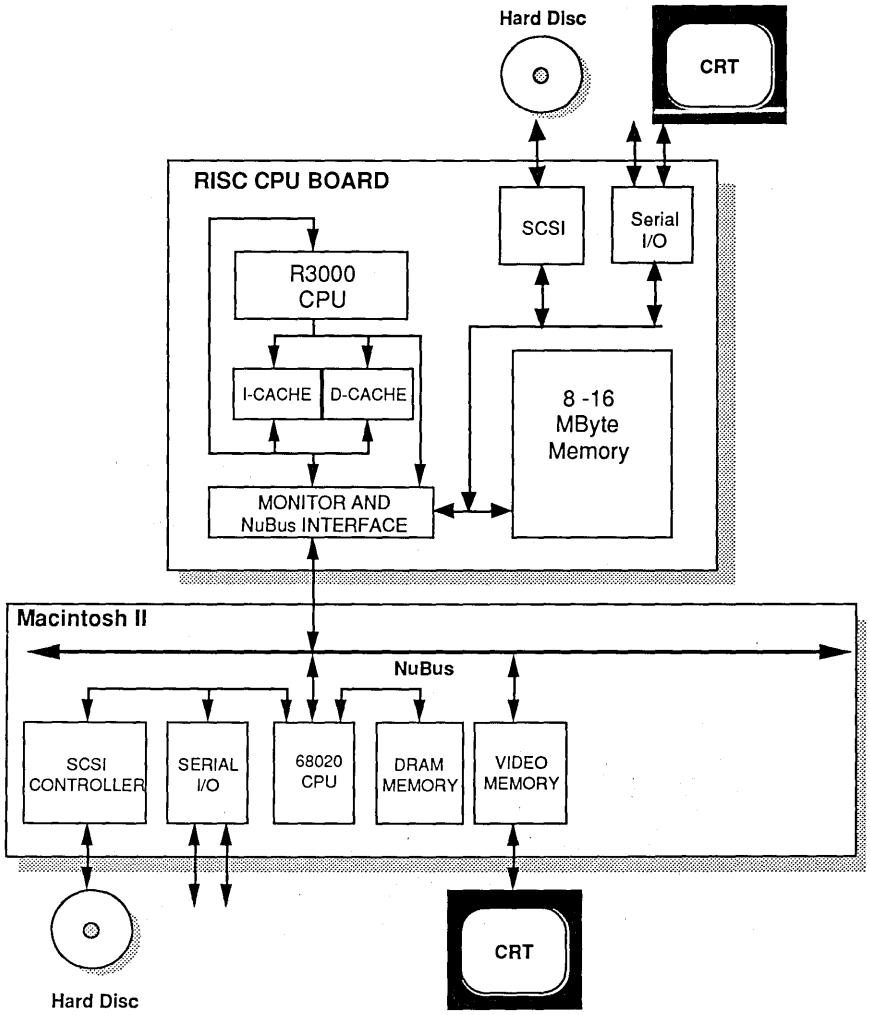
The CPU card is a 25 MHz R3000 system, using the Floating Point Accelerator and 64 KB each of instruction and data caches. The necessary hardware to run UNIX is also on the board. EPROMs on the card contain IDT's System Integration Manager (IDT/sim) which provides many debug and control features at the monitor level. The board can communicate with all Macintosh I/O and with Ethernet via the Macintosh NuBus. Additionally, the board has its own SCSI and serial I/O ports. A separate terminal can be used on the serial I/O port to just run command-line Unix, and an independent Macintosh compatible hard disc can be used on the SCSI port to provide faster disc access to the RISC machine by avoiding delays inherent in the NuBus and Mac O/S.

Main Memory

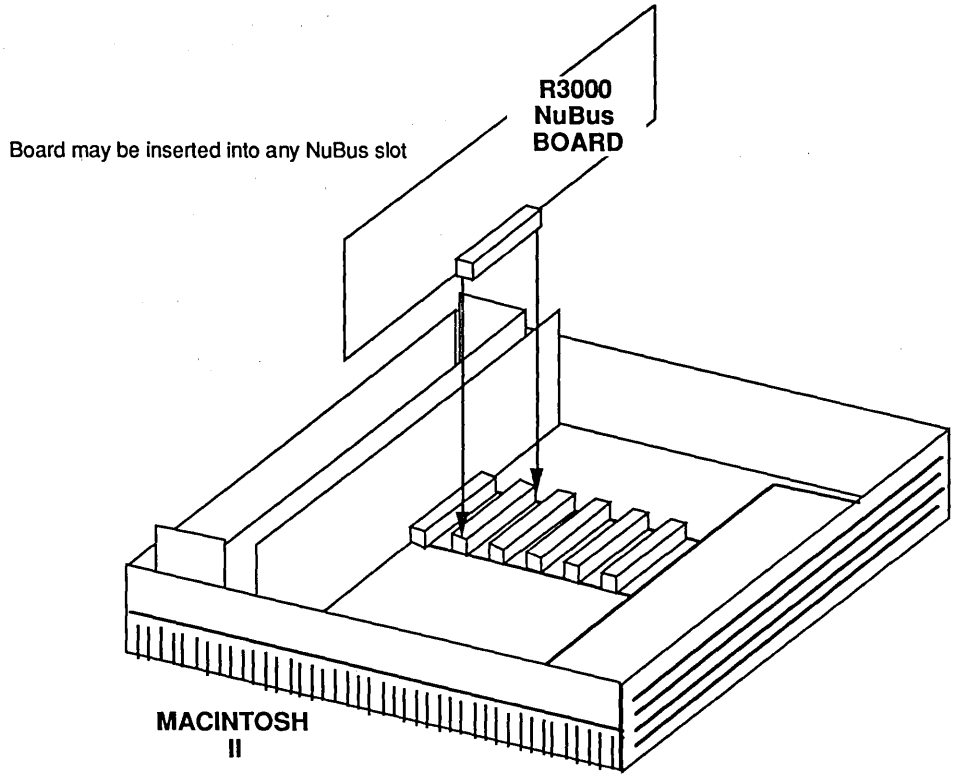
The MacStation 3 is available with either 8 or 16 MB of DRAM. This permits reasonably large programs to execute without excessive disc swapping.

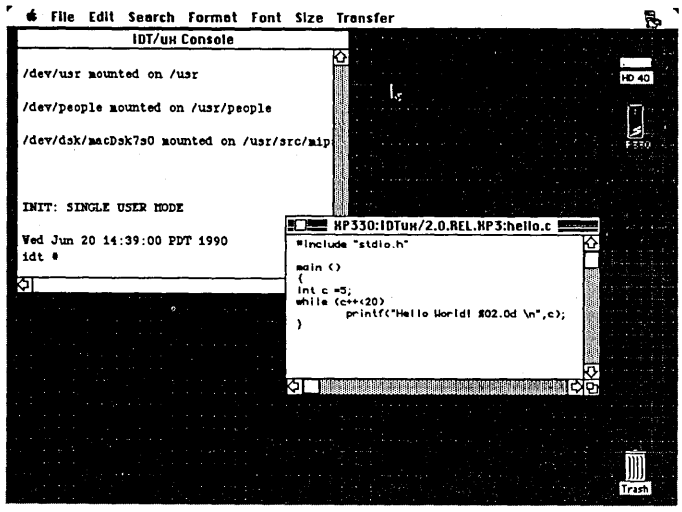
MacStation 3 SOFTWARE

The MacStation is shipped with all the software on tapes. It may be ordered with a 160 Mb hard disc with the software pre-installed, but tapes are still included for backup.



8





Screen Shot of Macintosh with IDT/ux running.

ORDERING INFORMATION

The MacStation 3 may be ordered in a variety of forms, ranging from the essential boards and software to complete systems with software pre-installed. The IDT/ux operating system requires a signed single-system license agreement which must be executed before the system can be shipped. Contact your IDT sales office for a sample of the license.

8 MB MacStation System 7RS503B8-L
Includes R3000 NuBus CPU card with 8 Mbyte RAM, IDT/ux. Requires a Macintosh II computer, system 6.01 or later, Apple tape drive, at least 160 Mbytes of free disc space.

16 MB MacStation System 7RS503B16-L
Includes R3000 NuBus CPU card with 16Mbyte RAM, IDT/ux. Requires a Macintosh II computer, system 6.01 or later, Apple tape drive, at least 160 Mbytes of free disc space.

MacStation Conversion Kit 7RS503TD8-L
Everything to convert a Macintosh II to a MacStation. Includes the 8 MB MacStation board, IDT/ux, a tape drive, a 160 Mbyte Hard Disc and cables.

Other Macintosh system configurations.
The MacStation can be supplied with any desired Macintosh configuration. Contact factory for a quotation.

IDT/ux Documentation

This is an eight-manual set of documentation for MIPS RISC/os (UNIX V 3 with BSD extensions). Included are: RISC/os Programmer's Reference Manual, User's Reference Manual, System Administration Reference Manual, System Administrator's Guide, Programmer's Guide, User's Guide, Streams Primer and Programmer's Guide, Guide to BSD on RISC/os.

IDT/ux Documentation Package 7RS551BDU

Other MIPS Software for MacStation

The following MIPS products are available for the MacStation. All require asigned license agreement prior to shipment. Contact your IDT sales office for a sample of the agreement.

SPP for MacStation 7RS992SMT-L
Source Code, site license, includes documentation

SPP/e for MacStation 7RS993SMT-L
Source Code, site license, includes documentation

8

Integrated Device Technology, Inc.

IDT/sim
SYSTEM INTEGRATION MANAGER
ROMable DEBUGGING KERNEL

IDT7RS901

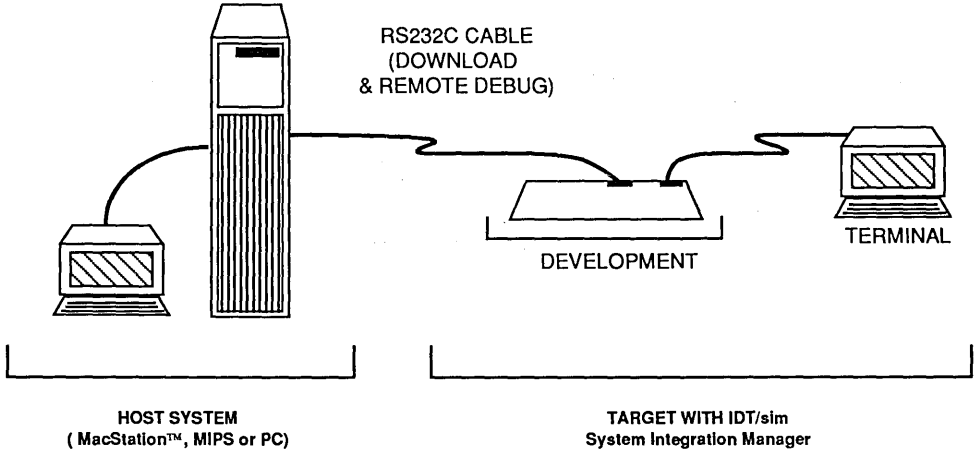
FEATURES:

- **Provides complete control over hardware and software for system integration**
- **Fits in 82Kb of EPROM space, plus 16Kb of RAM**
- **Provides CPU control for register and memory manipulation, cache access, and TLB management**
- **Includes standard I/O support**
- **Easy to add new commands and new I/O drivers.**
- **Complete support for MIPS symbolic debuggers. No additional code required**
- **Supports downloading code in either ASCII or binary formats**

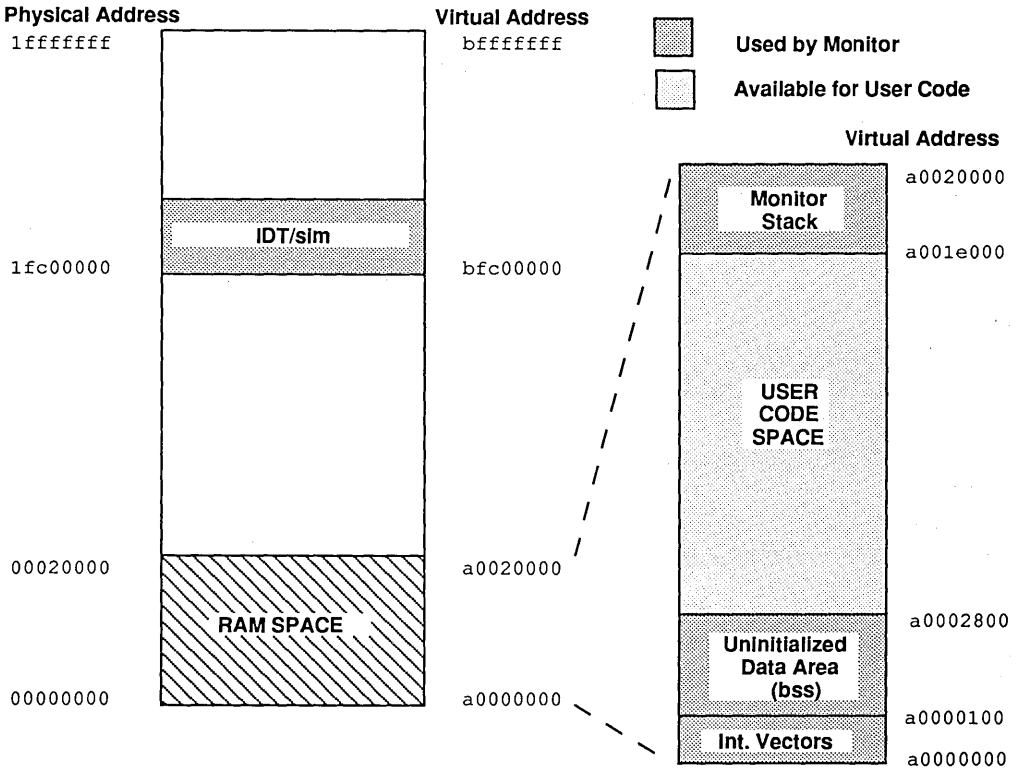
POWERFUL TOOL FOR R3000 SOFTWARE/ HARDWARE INTEGRATION:

The IDT7RS901 System Integration Manager (IDT/sim) is a ROMable software product that permits convenient control and debug of RISC systems built around the MIPS R3000 architecture. It permits users to quickly develop and debug stand-alone systems. Facilities are included to operate the CPU under controlled conditions, examining and altering the contents of memory, manipulating and controlling R3000 resources (such as cache, TLB and coprocessors), loading programs from host machines, and controlling the path of execution of loaded programs. Remote (source/symbolic) debugging is also supported.

IDT/sim requires 82Kb of EPROM space for code and data and 16Kb of ram sapce for uninitialized variable data and stack. The minimal I/O system supported uses UARTS. The default drivers support the 2681 or 68681 devices. Other devices can be added easily.



Debug Commands	Provides commands for CPU control such as execution, register and memory manipulation, cache access and TLB management.
Run Time Support	IDT/sim provides numerous entry points that client programs may access to perform standard operations. These include I/O support (open, close, read, write etc.), standard console I/O support (get/put char/string, printf) and facilities to install new commands and device drivers.
Remote Debug	Complete remote (source/symbolic) debug support is provided for operation with MIPS symbolic debuggers (DBX). This feature is incorporated into the monitor residing in EPROM so the user does not have to download any additional code (ie., dbgmon) to debug code running on his target system. Additionally, an easy to use interface is provided so user code that is interrupt driven may be debugged.
Download Support	Supports downloading code from a host to the users target machine. Both ASCII and binary formats are supported.



8

LIST OF COMMANDS

rad [-o|-d|-h]
Set the default radix to the requested base.

seg [-0|-1|-2|-u]
Set the default segment to the requested k-segment.

history|h
Displays the last 8 commands entered

help|? [commandlist]
Prints a list of the commands available in the monitor.

regsel|rs [-c|-h]
Selects display format for register names.

checksum|cs
Display the checksums for each of the 'EPROMs'.

init|i
Initialize prom monitor (warm reset)

dbgint|di [-e|-d>] <DEV>
Debug interrupt enable/disable - allows 'break key' to gen extr. int.

fill|f [-w|-h|-b|-l|-r] <RANGE> [value_list.]
Fills memory specified by range with value_list..

sub [-w|-h|-b|-l|-r] <address>
This command allows the user to examine and change memory interactively.

dump|d [-w|-h] <RANGE>
Display contents of memory.

move|m [-w|-b|-h] <RANGE> <destination>
Move the block of memory

compare|cp [-w|-b|-h] <RANGE> <destination>
Compare the block of memory

search|sr [-w|-b|-h] <RANGE> <value> [mask]
Search the area of memory for a value.

wc [-i] [-w|-b|-h] <RANGE> [value._list]
Fill cache memory with a pattern.

cacheflush/cf [-i|-d]
Flush both the i-cache and the d-cache.

rc [-i] <-w|-b|-h] <RANGE>
Display cache memory.

fr <reg#|name> <value>
Put <value> into the register

dr [reg#|name]
Print out the current contents of registers.

dis <RANGE>
Disassemble the contents of memory.

tlbdump|td [RANGE]
Dumps the contents of the translation buffer.

tlbflush|tf [RANGE]
Flushes the contents of the translation buffer.

tlbmap|tm [-i index] [-ndgv] <vaddress> <paddress>
Specify virtual to physical mapping in the translation buffer.

tlbpid|ti [pid]
Displays the current process identifier (pid).

tlbptov|tp <physaddr>
Search the translation buffer for translations which map to <physaddr>.

load|l <_device> [format]
Down load from an I/O device.

debug|db [DEV]
Enter remote symbolic debug mode with host.

go|g [-n] <address>
Begin execution at address

gotill|gt <address>
Continue execution from the current PC till break address encountered

call|ca <address> [arg1 arg2 ... arg8}
Invoke a 'C' language subroutine.

step|s [<count>]
Execute a specified number of instructions.

cont|c
Continues execution of the client process from where it last halted execution.

brk|b [addresslist]
Display currently set breakpoints or set break point at each of the addresses.

unbrk|ub <bpnumlist>
Unset breakpoints listed.

LIST OF RUN TIME SUPPORT ENTRY POINTS

__exit()
Returns control to the monitor.

_atob(str,intptr,base,seg)
Converts an ascii string to an integer.

_clear_cache(begin_addr,num_bytes)
Clears a selected area in I and D cache

_cli(cmd_table,prompt)
General purpose command line interpreter

_flush_cache()
Flushes both the I and D cache.

_get_range(str,start,end)
Parses the range specification.

_getchar()
Get a character from the standard input device.

_gets(str)
Get a string from the standard input device.

_help(argc,argv,cmd_table)
Print the usage line for all specified commands

_install_commands(cmd_table)
Allows the user to extend the command set of the standard monitor.

_install_immediate_int(ptr_user_int_rt)
Installs a pointer to a user interrupt function that will be called by the monitor when an exception/ interrupt occurs.

_nstall_new_dev(dt_ptr,diptr)
Installs a new device driver that will be recognized by IDT/sim

_install_normal_int(ptr_user_int_rt)
Installs a pointer to a user interrupt function that will be called by the monitor when an exception/ interrupt occurs.

_ioctl(fd,cmd,arg)
Sets flags for i/o characteristics and/or calls driver ioctl routines.

_open(device,flags)
Opens a device for reading and/or writing.

_printf(format,[args])
Formatted print routine

_putchar(c)
Output a character to the standard output device.

_puts(str)
Output a string to the standard output device.

_read(fd,buf,cnt)
Read data from an external device.

_reset()
Resets the monitor

_setjmp(cur_cntx)
Save the current context so that non-local goto's may be implemented.

_longjmp(cur_cntx)
Restores the saved context so that non-local goto's may be implemented.

_showchar(c)
Prints the character passed to it in a visible manner

_strcat(s,t)
Concatenate two strings (39)

_strcmp(s,t)
Compare two strings (36)

_strcpy(s,t)
Copy one sting to another (38)

_strlen(s)
Determine the number of characters in a string. (37)

_tokenize(cmdline,argv)
Parse command line and build argc/argv structure

_write(fd,buf,cnt)
Write data to an external device.

ORDERING INFORMATION

To order the IDT System Integration Manager, order the Developmental Use License AND order the software on the appropriate media. The license will be shipped to you for signature; on return the software will be shipped. You may also order binary distribution rights for the run-time version of the monitor. Ask your IDT sales office for information.

Licenses

Developmental Use License **7RS901SLV**
Permits purchase of up to six copies of source code (any media combination) and use of source code to develop run-time binaries on up to six machines at a time, but does not permit inclusion of the run time code in an end product.

Binary Distribution Rights **7RS901BLP-L**
Extension to Developmental Use License to permit inclusion of binary code into end product. Development Use License must be referenced on order or ordered simultaneously. This license permits up to 100 copies to be distributed royalty-free. Additional copies are subject to the royalty below, or a one-time buyout.

Binary Distribution Sublicense **7RS901BLC-L**
Per Copy Royalty for distribution of runtimes developed using the System Integration Manager beyond the first 100..

Maintenance Agreement **7RS901SSY**
One year free minor updates, and discounted upgrade to major update versions. We supply a direct telephone contact for support.

Source Media

Source for 286/386, MS-DOS **7RS901SAF-L**
Use with IDT/c C-Compiler (7RS903). Shipped with both 1.2 MB 5.25" and 1.44 MB 3.5" diskettes

Source for 286/386 PC, SCO Xenix **7RS901SXX-L**
Use with IDT/c C-Compiler (7RS903). Developmental Use License number must be referenced on order, or must be ordered simultaneously.

Source for IDT MacStation, on Mac Disc **7RS901SMD-L**
Use with MIPS C Compiler supplied with MacStation or with IDT/c. Developmental Use License number must be referenced on order, or must be ordered simultaneously.

Source for MIPS or SUN Machine, QIC-24 TAR Tape **7RS901SUU-L**
Use with MIPS C Compiler or with IDT/c. Developmental Use License number must be referenced on order, or must be ordered simultaneously.

EPROM Versions

The following versions of IDT/sim are supplied in EPROMs for the indicated hardware. These versions are for updating the hardware to the latest version of the monitor.

For Evaluation boards and Prototyping Systems **7RS901BAP**
Use with 7RS382,383, or any 7RS300 series module Prototyping Platform

For the MacStation 1 ..... **7RS901BBP**
Use with 7RS501 Original MacStation CPU board

For the MacStation 2 ..... **7RS901BCP**
Use with the 7RS502 MacStation CPU board

Auxiliary Download Programs

For downloading code from a MIPS machine into IDT/sim. This software includes programs to convert MIPS object code into S-records and to download either ASCII or binary S-records to a remote target. This software is only needed with MIPS computers; all other machines (including the MacStation) have standard utilities available to perform this function.

Integrated Device Technology, Inc.

IDT/c
MULTI-HOST C-COMPILER SYSTEM

IDT7RS903

FEATURES:

- **Includes C-compiler, Optimizing Scheduler, Assembler, and Linker**
- **Optional Floating Point Software**
- **Meets Plum Hall 2.00 ANSI C validation suite**
- **Runs on 80286 and 80386 machines under MS-DOS™ or XENIX™, on MIPS machines under RISC/os, and on MacStation™ under IDT/ux**
- **Supports entire IDT family of MIPS ISA Processors: R3000, R3001, R3051, and R3052**
- **Loader communicates with IDT's System Integration Manager (IDT/sim)**
- **Provides control over multiple memory segments**

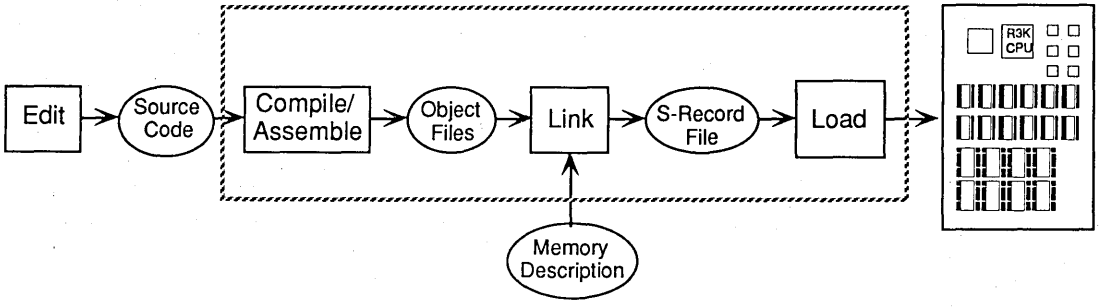
OPTIMIZING C-COMPILER SYSTEM:

IDT/c consists of a set of software products that run on a variety of platforms, and which together produce highly efficient code for R3000 CPUs. The code can be downloaded in several formats to a target machine for execution. On the target machine, the code can be controlled with IDT's System Integration Manager (IDT/sim).

The compiler is based on the popular GNU C compiler, and is fully compliant with ANSI C.

The entire package is available for execution on 286 or 386 machines under MS-DOS or XENIX, as well as the MIPS and SUN workstations, and IDT's MacStation single user workstation.

For any platform, IDT/c can be ordered with or without a software floating point library. A switch in the compiler determines if floating point instructions will result in R3010 instructions in the object code or whether calls to the floating point library will be made instead.



IDT/c System Flow

8

DESCRIPTION:

The IDT/c C-Compiler System is a complete development package for CPUs based on the R3000 architecture. It contains an optimizing cross compiler, scheduler optimizer, cross assembler, linker, and a downloader. The 'C' compiler is compliant with ANSI 'C' standard and performs the optimizations available in state of the art 'C' compilers. The assembler supports the R3000 machine instructions and architecture described in the book by Gerry Kane, "MIPS RISC Architecture", including both native and synthetic instructions. The complete IDT/c package runs on a variety of host machines and operating systems and is part of IDT's cross development system tools which include other packages such as debug monitors and libraries.

Compiler

The C pre-processor is GNU cpp and the compiler itself is based on GNU C. All C-preprocessing features are supported. The combination of the compiler and assembler included in IDT/c has been tested for compliance to the ANSI 'C' standard using the Plum Hall test suite and is compliant. C programs written for the MIPS C compiler may also be compiled without modification.

The C compiler performs extensive optimization in multiple passes through the code. Each of the many optimization techniques can be individually switched on or off with compiler directives.

Optimizing Scheduler and Assembler

The IDT cross assembler input is compatible with source code written for the MIPS assembler. It implements the R3000 native instruction set as well as the augmented synthetic instructions defined in the "MIPS RISC ARCHITECTURE" book by Gerry Kane. There are some extensions in the IDT cross assembler that provide the programmer with more control over code generation, such as 'laiu' - load address upper and 'oria' - load address lower, enabling direct programming in pure assembly language. The assembler produces .o files which are later linked together with other files to produce an executable file.

The scheduler first expands the synthetic instructions into the native instruction set. It then rearranges code to allow for and take advantage of R3000 pipeline architecture. At the same time the scheduler analyzes loads of static constants and makes use of previously loaded constants that are close in value.

Memory description file

The memory description file is used to instruct the linker where to place object modules in the R3000 memory map. It tells linker what address classes are legal, what addresses exist within those classes, and what addresses should be written to output files. The file consists of a sequence of class specifications (CODE, DATA, etc.) and associated address ranges.

Linker

The linker combines together separately assembled program files into one object module. Command line switches may be used to override the memory description file.

The format of object code produced by the assembler in IDT/c is not compatible with the format produced by the MIPS assembler, so modules compiled by the MIPS software cannot be linked directly with modules compiled by IDT/c. Recompilation under IDT/c is required.

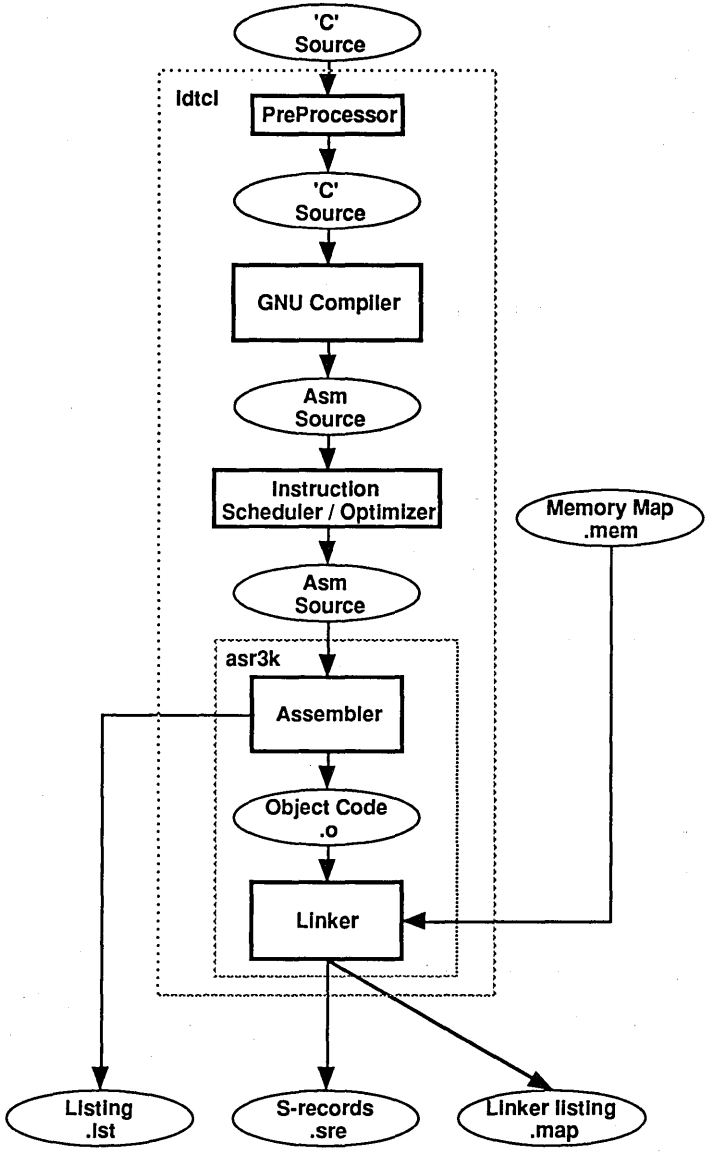
There are three types of output file formats supported: S-Records, INTEL hex, and binary image. The S-Record files are useful in down-loading to target boards. The INTEL hex format files are useful for EPROM programing because the linker provides for the code to be divided into multiple files under this format.

Endianness

IDT/c is Big-Endian.

Floating Point Library

IDT/c may be ordered with a floating point library. A switch in the compiler is set at compile time to determine how the compiler should handle floating point instructions. In the the normal mode, it will produce R3010 Floating Point Accelerator instructions in the object code. If the switch is set the other way, the compiler will insert calls to the floating point library instead, and the floating point library must be lavailable at link time. Because the compiler knows about the libary during compile time, it can perform optimizations not otherwise possible and keep the execution penalty for using software instead of hardware to about a factor of 4 in very fp intensive code.


'C' Source
Idtcl
PreProcessor
'C' Source
GNU Compiler
Asm Source
Instruction Scheduler / Optimizer
Asm Source
asr3k
Assembler
Object Code .o
Linker
Memory Map .mem
Listing .lst
S-records .sre
Linker listing .map

IDT/c Flow

OPTIMIZATION PASSES

Multiple optimization passes are performed by the GCC compiler. Below is a brief description of what takes place on each pass. Note that switches can be used in the compiler to turn individual optimization choices off or on, providing the programmer with a great deal of control over how the compiler modifies the code.

Jump optimization

Simplifies jumps to the following instruction, jumps across jumps, and jumps to jumps; while deleting unreferenced labels and unreachable code.

Register Scan and common subexpression elimination

Finds first and last use of each register for purposes of subexpression elimination while performing constant propagation.

Loop optimization and strength reduction

Moves constant expression code outside of dynamic loop.

Data flow analysis

The program is divided into basic blocks and identifies the life of values in registers. Once done, then code producing unused results can be eliminated and unreachable loops are eliminated.

Local register allocation

Allocates registers to be used inside each basic block.

Global register allocation

Assigns registers for values which live across basic block boundaries.

Final Pass

The final pass is to generate assembler code. At this point peephole optimizations are performed as well as generating and optimizing the function entry and exit code sequences.

PERFORMANCE COMPARISONS

Execution

To obtain a measure of the efficiency of the IDT/c compiler, a set of benchmark programs was compiled under both IDT/c and the MIPS compiler, and the size and execution time of the resulting binaries were compared.

Execution Time Comparison

	Code Size	Exec. Time
Compiled with MIPS C	1.0	1.0
Compiled with IDT/c	1.20	1.19

Compile Time

The time required to compile a program under IDT/c depends on the machine speed, type, and configuration. For comparative purposes, the Stanford benchmark was compiled under a variety of hosts and the results are shown below. For reference, the same program was also compiled using the MIPS compiler.

Compile Time Comparisons

Host	Compile Time
MIPS C on MIPS Machine	24 sec.
IDT/c on MIPS Machine	25 sec.
IDT/c on 10 MHz 286, MS-DOS	695 sec.
IDT/c on 25 MHz 386, Xenix	70 sec.

COMMAND LINE SWITCHES

-E :
pre-process only. .S file is expected. Pre-processed file is written to the standard output.

-O :
Optimize (GNU cc -O option).

-O1 :
Optimize even more (GNU cc options: -fstrength-reduce -fforce-addr -fforce-mem -fcombine-regs -finline-functions).

-c :
Assemble only, do not link. Expected are filenames with .s or .S suffixes. Output files (in absence of -o) will have .o suffix.

-ZA :
produce assembly listing.

-o xxx :
Name output file. The default output name is 'out.sre'.

-ZL :
Produce link map.

-Fxxx.xxx :
Use xxx.xxx as memory layout description file. In absence of -F option the default is to use file idt.mem in default library directory.

-ZThhhhhhhh :
Specify text loading address, hhhhhhhh is address in hex, up to 8 hex digits. This will override .mem file definitions.

-ZDhhhhhhhh :
Specify data loading address, hhhhhhhh is address in hex, up to 8 hex digits. This will override .mem file definitions.

-e name :
Use global 'name' as program start address.

-noenv :
Do not include default library modules which define the order of program sections and global symbols that point to beginning and end of text, data and bss.

-nostdlib :
Do not include library for linking with IDT PROM monitor.

ASSEMBLER DIRECTIVES

.align n, n=1-3
align so that n least significant bits of address are 0.

.ascii "string"
assemble string.

.bss
store following into bss section.

.byte arg,arg,...,arg
assemble arguments into consecutive bytes.

.data
store following into data section.

.end name
Included for MIPS asm compatibility but ignored.

.ent name
Included for MIPS asm compatibility but ignored.

.extern name,n
Import symbol 'name' that refers to n bytes of storage . (included for MIPS asm compatibility and ignored).

.globl name
export defined symbol.

.half arg,arg,...,arg
assemble arguments into consecutive halfwords.

.set argument
argument can be :

at - error flag every use of $1.

noat - disable errors due to user's usage of $1 (at).

reorder - enable scheduling to resolve pipeline conflicts.

noreorder - disable scheduling.

.space n
skip next n bytes, advancing location counter by n.

.text
store following into text section.

.word arg,arg,...,arg
assemble arguments into consecutive words.

SEGMENT
The SEGMENT directive selects the address segment where the following code or data will be stored. It is used to implement '.text', '.data' and '.bss' which are MIPS compatible segments. Using this directive the user can create other custom segments.

8

ORDERING INFORMATION

The IDT/c C-Compiler is an efficient R3000 C-compiler system based on the popular GNU C and hosted on a variety of computers. The IDT/c system includes the compiler, assembler, scheduler and linker. All PC versions of the software are shipped with both 1.2 MB floppy discs and 1.44MB 3.5" diskettes. A "boxtop" single user license is included with the product.

Media, without Floating Point

The software listed below does not include the floating point library.

For 286 machine, MS-DOS 7RS903BAF-N
Not recommended for large, complex programs. At least 2MB RAM recommended. Requires DoOS version 3.3 or greater.

For 386 machine, MS-DOS 7RS903BBF-N
Note: as of 10/1/90, this product is the same as the 286 MS-DOS version, but a performance enhancement is planned for early 1991. Registered users of this version of the software will automatically receive the enhanced software as soon as available.

For 286 machine, SCO Xenix 7RS903BYX-N
For 386 machine, SCO Xenix 7RS903BXX-N
For MIPS machine RISC/os, on QIC-24 TAR Tape 7RS903BUU-N
For MacStation, on Macintosh Tape 7RS903BMD-N
Runs on MacStation R3000 board under IDT/ux.

For SUN Sparcstation, on QIC-24 TAR tape 7RS903BWU-N

Media, with Floating Point Library

The software listed below includes the floating point library.

For 286 machine, MS-DOS 7RS903FBAF-N
Not recommended for large, complex programs. At least 2MB RAM recommended. Requires DoOS version 3.3 or greater.

For 386 machine, MS-DOS 7RS903FBBF-N
Note: as of 10/1/90, this product is the same as the 286 MS-DOS version, but a performance enhancement is planned for early 1991. Registered users of this version of the software will automatically receive the enhanced software as soon as available.

For 286 machine, SCO Xenix 7RS903FBYX-N
For 386 machine, SCO Xenix 7RS903FBXX-N
For MIPS machine RISC/os, on QIC-24 TAR Tape 7RS903FBUU-N
For MacStation, on Macintosh Tape 7RS903FBMD-N
Runs on MacStation R3000 board under IDT/ux.

For SUN Sparcstation, on QIC-24 TAR tape 7RS903FBWU-N

Floating Point Upgrade

The version of the compiler without floating point may be upgraded to add the floating point library. To upgrade, contactyour IDT sales office. Indicate the order code and serial number for your original software on your order, so we will ship the correct format.

Floating Point Upgrade 7RS905BZU

Maintenance

Maintenance 7RS903BSY
Includes free upgrades for one year and direct telephone contact for support

Integrated Device Technology, Inc.

CROSS ASSEMBLER FOR IBM PCs AND CLONES

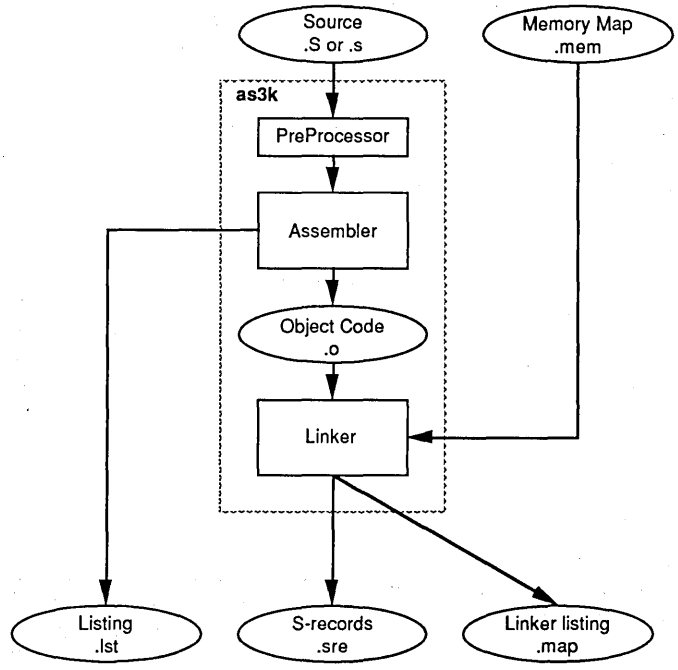
IDT7RS904

FEATURES:

- **C-Like Pre-processor allows programmer to extend Instruction set**
- **Provides for separate code segments loaded at different memory locations**
- **Outputs S-records and Intel HEX**
- **Provides symbolic and hex instruction listing**
- **Lists Absolute addresses of code segments and symbols**
- **Interfaces to IDT System Integration Manager**

DESCRIPTION:

IDT7RS904 is a cross assembler with a C-like pre-processor and linker that produces Motorola S-record or INTEL HEX downloadable files and a downloader. It is intended for cross-development with R3000 as target architecture. The assembler is compatible with files written for the MIPS assembler. The assembler supports the R3000 machine instructions and architecture descibed in the book by Gerry Kane, "MIPS RISC Architecture". The cross assembler package runs on variety of host machines and operating systems. The pre-processor, assembler and linker are invoked by a single driver named "asr3k".



Assembler Flow

8

PRODUCT FEATURES

Pre-processor

The C-like pre-processor is provided to process assembly language files such that the programmer can extend the instruction set of the cross assembler. All C- preprocessing features are supported.

Assembler

Assemblers for R3000 differ from 'classical' assemblers because they attempt to transparently provide the richer instruction set. This is due to the RISC architecture, which provides only essential instructions and achieves high performance in return. In order to enable higher efficiency in assembly-language programming, assemblers introduce instructions that do not directly map into the machines instruction set. In this way programmers in effect can program with a richer instruction set than that of the processor.

The 7RS904 is compatible with the MIPS assembler. It implements the R3000 instruction set as defined in the "MIPS RISC ARCHITECTURE" book by GERRY KANE (appendix D). With a few exceptions programs written for the MIPS assembler will directly compile. The cross assembler varies from the MIPS assembler in that it requires labels, if present, to start in column 1 and operators to start in a column greater than 1.

Linker

The linker combines together, seperately assembled program files into one object module. It uses information from the memory description file to determine where the object module will be placed in the R3000 address map. Command line switches may be used to override memory description file.

Memory DescriptionFile

The memory description file tells linker what address classes are legal, what addresses exist within those classes, and what addresses should be written to output files. The memory description file consists of a sequence of class specifications (CODE, DATA, etc.) and associated address ranges. In this way the linker can be instructed to place code at different locations.

The linker can generate three different types of output files. The file type is controlled from the memory desricption file, by placing a FILETYPE specification in the output file specification line. There are three choices: DUMP - Generates single .sre S-record file. (default), INTEL - Intel hex file format, BINARY - Generates a binary image of the file.

DownloadableFiles

The linker can output S-record files (.sre), in which individual S- records are lines of text as interpreted by the host system conventions, ie. they are separated by line delimiters. The 'dl' program can convert files to S-records with no intervening delimiters.

EPROM Files

The linker can output INTEL hex files for the purposes of burning EPROMs. The linker may be instructed to divide the output code and data segments into multiple files to accomodate seperate sockets and address spaces.

Extentions

There are some extensions in the cross assembler that provide the programmer with more control over code generation, such as 'lau' - load address upper and 'lal' - load address lower, enabling direct programming in pure assembly language.

lal reg,addr	Load Address Lower. Loads (using ori) lower 16 bits of address into register. Address is relocatable symbol (can be external).
lau reg,addr	Load Address Upper. Loads (using lui) upper 16 bits of address into register. The lower 16 bits are cleared. Address is relocatable symbol (can be external).

ORDERING INFORMATION

7RS904 Cross Assembler for PCs

This is a macro-assembler that runs on any PC running MS-DOS. A single user license is included.

For PC/XT, MS-DOS, on 320 Kb Floppy	7RS904BPL-N
For PC/AT, MS-DOS, on 1.2 Mb Floppy	7RS904BAF-N

Integrated Device Technology, Inc.

IDT/fp FLOATING POINT LIBRARY FOR USE WITH R3000 COMPILERS

ADVANCE INFORMATION
IDT7RS905

FEATURES:

- **Conforms to IEEE 754 format**
- **Converts ASCII to/from Floating Point**
- **Converts Integer to/from Floating Point**
- **Adheres to IEEE 754 Error Handling**
- **Works with IDT/c on all platforms**

DESCRIPTION:

The IDT7RS905 product is a floating point (IEEE-754) arithmetic library for use in IDT79R3000 systems that do not incorporate a IDT79R3010 floating point coprocessor. It provides the basic single and double precision arithmetic functions (add, subtract, multiply and divide) as well conversion routines between different precisions, integer and ascii formats. The IEEE-754 single precision floating point format represents numbers ranging from ±1.2E-38 to ±3.4E+38 with 24 bit mantissa precision. The double precision format offers a range of ±2.2D-308 to ±1.8D308 with a 53 bit mantissa. The accuracy of the floating point library is within one least significant bit. The IEEE floating point format defines special representations for underflow (result = zero), overflow (result = + INF or - INF), and invalid operation (result = Not a Number, NaN). The floating point library adheres to the IEEE-754 error handling procedure in all applicable cases.

SUPPORTED OPERATIONS

Addition	FPADD(a,b) & DPADD(a,b)
Subtraction	FPSUB(a,b) & DPSUB(a,b)
Multiplication	FPMUL(a,b) & DPMUL(a,b)
Division	FPDIV(a,b) & DPDIV(a,b)
Comparision	FPCMP(a,b) & DPCMP(a,b)
Integer to FP	FPFLT(int) & DPFLT(int)
FP to Integer	FPINT(sp) & DPINT(dp)
DP to SP	DPTOSP(dp)
SP to DP	SPTODP(sp)
ACSII to FP	FASCBIN() & DASCBIN()
FP to ASCII	FBINASC() & DBINASC()

8

ORDERING INFORMATION

This software contains a library of floating point routines that perform functions equivalent to the R3010 Floating Point Accelerator chip. The routines can be linked with output from any of the compilers listed below, and are ROMable. The execution time is roughly 30 times that required by the R3010. The Developmental Use License and copies of the software in one or more media should be ordered separately. The license must be signed before the software is shipped.

Developmental Use License **7RS905BLV**
Permits purchase of up to six copies of the software in any media, and permits up to six copies running in developmental machines, but does not permit inclusion of code in an end product.

Binary Distribution Right **7RS905BLP-L**
Permits unlimited, royalty free distribution of run-times containing the Floating Point Library .

For use with 7RS903BAF-N (PC) **7RS905BAF-L**
For use with MacStation **7RS905BMD-N**
For use with MIPS machine, RISC/os **7RS905BUU-L**
For use with 7RS903BXX
(386 Xenix Disc) **7RS905BXX-L**
For use with 7RS903BYX
(286 Xenix Disc) **7RS905BYX-L**
Binary Distribution Rights **7RS905BLP**
Permits royalty-free distribution of run-time code.

Maintenance **7RS905BSY**
One year free updates

Integrated Device Technology, Inc.

THIRD-PARTY DEVELOPMENT TOOLS AND APPLICATIONS SOFTWARE FOR IDT RISC PROCESSORS

OVERVIEW

The MIPS/IDT RISC Microprocessor family is supported by a wide variety of third-party development tools and applications software. Many of these tools are software products, useful across the entire line of processors; others of these are hardware development tools, appropriate for one or two members fo the family.

As the MIPS architecture is increasingly popular and successful, many new tools are constantly being announced. IDT encourages our customers to work closely with their local sales representative for a current list of third party support. This listing is intended to be reasonably current as of the data of this document.

Type	Vendor	Product Name	Phone
Logic Analyzer			
	Arium	ML4400	(714)-978-9531 (800) 862-7486
	Biomation	CLAS4000	(800) 538-9320 (408) 988-6800
	Hewlett Packard		local HP sales office
	Tektronix	DAS9200	(800) 426-2200 (503) 627-7111
In-Circuit Emulator			
	Embedded Performance		(408) 980-8833
Hardware Simulation Models			
	Mentor Graphics	HML	(800) 547-7390 (503) 626-7000
	Logic Modeling Systems		(408) 957-5276
Software Simulation Models			
	Logic Automation	R4000 Sim	(503) 690-6900
Ada Development on Vax host			
	InterACT		(212) 696-3700
	Telesoft		(619) 457-2700
VME Board			
	Creative Electronics, Switzerland		(022) 792 57 45 (022) 792 57 48 (fax)
	Lockheed Sanders, Inc.		(603) 885-6022
	Omnibyte		(708) 231-6880
Real-time Operating Systems			
	JMI	C-EXECUTIVE™	(215) 628-0846
	Lynx OS	Lynx™	(408) 354-7770
	Wind River Sys.	VxWorks™	(415) 748-4100
Laser Printer Languages			
	Adobe Systems	Postscript™ PDL	(415) 961-4400
	Microsoft	TrueImage™	(206) 882-8080
	Peerless Group	PeerlessPage™ OS	(213) 536 0908

8

TABLE OF CONTENTS

9

Integrated Device Technology, Inc.

RISC AND THE MEMORY HIERARCHY

APPLICATION NOTE AN-19

By Michael J. Miller

The goal of every computer architecture is to decrease the time it takes to execute a task. With RISC the goal is no different, but the way this is achieved is different than previous architectures such as CISC. To achieve this desired goal, many aspects of CPU design must be addressed. One of the techniques used in RISC to achieve a faster execution rate is to implement a reduced instruction set to gain shorter clock cycles. However it is not sufficient to just speed up the cycle time of the processor. One must pay attention to how to best feed the streamlined processor. In other words, the architect needs to solve the problem of how to support the increased instruction and data bandwidth required by the CPU. Beyond speeding up the datapath, the architect must also address issues such as how to achieve efficient exception handling, fast context switching, memory management and fast Floating Point operation. For further performance,the architect must also address how to tie together multiple copies of this architecture to achieve ever increasing power. This application note will concentrate on how these issues are addressed by the IDT79R3000.

The Performance Equation

The measurement of performance is the time it takes to complete a task, (which is a product of the number of instructions to be executed for the task, the number of cycles per each instruction and cycle time (Figure 1.)) The architects of some of the most recent generation of CPUs (CISC) have spent most of their time addressing the number of instructions per task. By increasing the complexity of the instruction set, the number of instructions to execute the task was decreased. However, each term of the basic performance equation is not independent from the others. In increasing the complexity of the instructions, the number of cycles to execute the instructions increased. The RISC philosophy is to roll-back the complexity of the instructions and reduce the cycles per instruction. This reapportions the resources of the silicon to execute more instructions in parallel as well as include structures to control cache and memory management.

$$\frac{\text{TIME}}{\text{TASK}} = \frac{\text{INSTRUCTIONS}}{\text{TASK}} \times \frac{\text{CYCLES}}{\text{INSTRUCTION}} \times \frac{\text{TIME}}{\text{CYCLE}}$$

Figure 1. Performance Equation

The first term of the performance equation is the number of instructions per task. Whether it be RISC or CISC, the number of instructions to be executed can be minimized by optimizing compilers that have come into their own in the last 6 years. RISC strives to take advantage of these optimizing compilers by including simple register to register instructions which can be utilized more efficiently by the compiler. Additionally, it takes less time to operate on data already in the register than it does to go external to the CPU. The RISC philosophy is to include a large number of registers on the CPU chip and to require the compiler to make efficient use of the registers through effective allocation. Register allocation is an operation that allows optimizing compilers to achieve some of the biggest gains in performance.

In choosing those instructions to be included in the architecture, the designers of RISC analyzed the typical programs executed on the CPU. It was found that about 90% of all instructions executed in a task were simple loads and stores, ALU operations and branches. If the architects could speed up these instructions, they could make a significant performance gain. The other 10% of instructions were more complex operations. Previous generations of CPUs used more of the silicon for this 10% than they used for the 90% of simple operations. It was found that few architectural additions could be made to speed up this 10% of the instruction space without slowing the execution of the other 90% of the instructions. Therefore, RISC design focuses in the area that promises the biggest gain, i.e. the simpler instructions.

Once the architects isolated the key instructions to speed up, they could minimize the cycles that it took to execute those instructions. Pipelining was employed so that on every clock cycle a new instruction could be started before the previous one completed. Although the goal of RISC is to execute one instruction on every clock cycle, not all instructions can be completed on one clock cycle. Some instructions have a latency effect. These instructions typically are branches, loads and stores. Therefore, the task is to remove the latency as much as possible in these instructions to achieve the single-cycle execution goal. These latencies are mainly caused by the external memory pipelines for loads and branches. When branching, the way the test conditions are determined has the most affect on the latency. Fast RISC CPUs therefore, only compare operands for equality or against zero to eliminate carry/propagate chains in ALUs. This solution simplifies the architecture for increased speed without compromising the power.

The third part of the equation is to maximize the clock speed. This is done through improving the process technology that is used to implement the architecture, (such as bipolar, NMOS or CMOS), as well as the implementation of the architecture in it's logical form. As the internal delays are decreased and clock cycles shortened, the impact of external paths becomes greater. With RISC, the critical path external to the CPU is through the memory. To speed up the memory access for systems with big main memories implemented in DRAM, caches have been employed. The access time of the cache is determined by the organization and the implementation of the tag checking. Therefore, caches are integral to supporting the increased clock speed of RISC in big systems, but the caches alone cannot keep the CPU supplied.

The Memory Hierarchy

Programs and data are stored in DRAMs and secondary storage devices like hard disks which have slow access times in comparison with a cycle time of the ALU. The total RISC system implementation utilizes a memory hierarchy where each level of the hierarchy is, in effect, a cache for the next level down. The top of the hierarchy is a very small fast cache (register file) while the bottom level (main memory) is slow and very large. As an example, to support 20 MIPS operation, the CPU needs at least 200 Mbytes/sec of instruction and data bandwidth while main memory typically only supports 28 Mbytes/sec (see Figure 2). Previous CPUs, like the 68020, only required 15 Mbytes/sec to support 1.5 MIPS of performance.

9

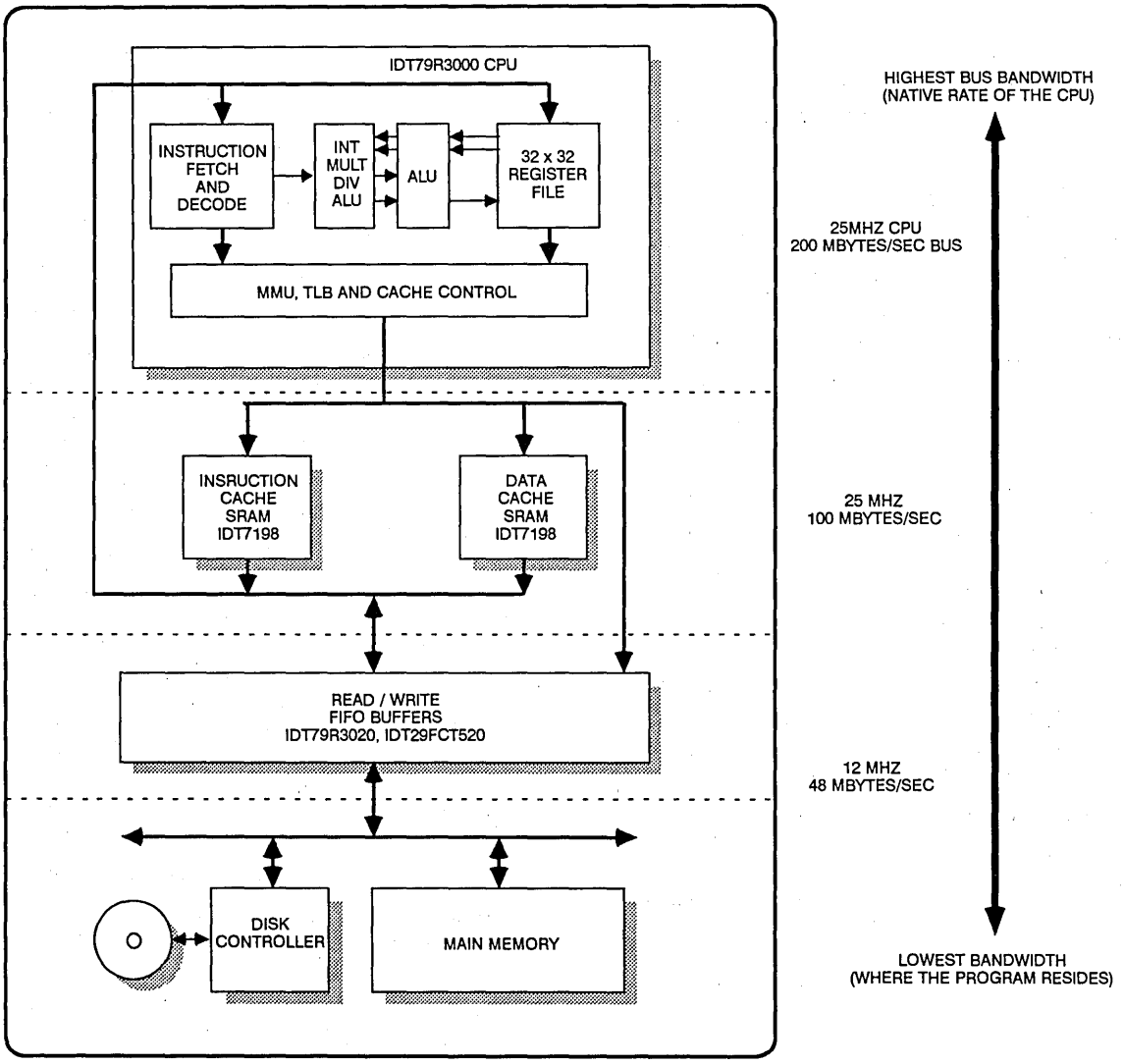


Figure 2. The Memory Hierarchy

The first level of the hierarchy is the register file that keeps the ALU fed with data. The management of the register file is performed at compile time. Without optimizing compilers, the total power of RISC cannot be realized.

The second level of the hierarchy is the instruction and data caches. Because RISC CPUs require a new instruction every clock cycle, the access time of the cache SRAM is the pacing delay in a RISC CPU's cycle time. To further increase the bandwidth into the RISC CPU, a separate cache for data is used.

The third level hierarchy is the main memory consisting of cache hard disks. Today's RISC CPUs employ on-chip a memory management unit (MMU) to manage the swapping of program segments between main memory and the disk. The Translation Lookaside Buffer (TLB) is a key element in the MMU that translates the logical addresses to the physical address of the programs stored in main memory. The TLB is a look-up table of addresses. Since circuitry to implement TLBs is costly in silicon area, TLBs are kept small and function as another cache for a large master look-up table kept in main memory.

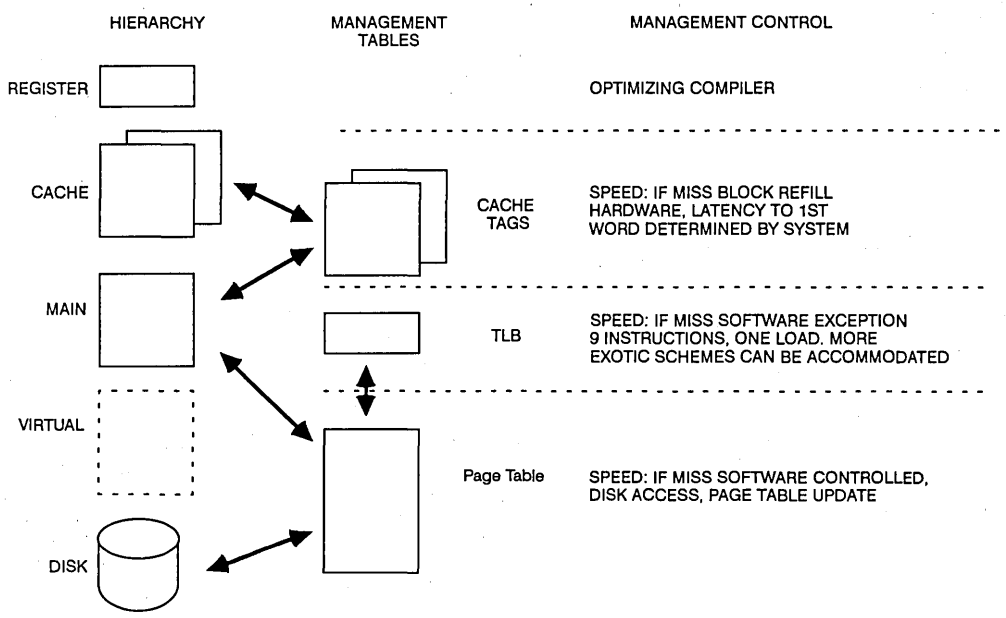


Figure 3. Managing the Memory Hierarchy

More than a Data Path

Therefore, RISC CPUs need not only a datapath, but also a memory hierarchy that must be managed by the CPU and software in order to operate efficiently. All levels of the hierarchy are used to feed the insatiable appetite of the instruction and data path of today's RISC CPUs. Since the reduced instruction set is only one aspect of the RISC CPU technique, perhaps a more appropriate name would be Bandwidth Increased Streamlined Computer (BISC). This new term could be used at the 'risc' of adding more letters to todays' already full bowl of alphabet soup!

9

Integrated Device Technology, Inc.

INTERRUPT LATENCY AND HANDLING IN THE IDT79R3000

APPLICATION NOTE AN-26

By Satyanarayana Simha

INTRODUCTION

The exception processing capability of the IDT79R3000 is provided to assure an orderly transfer of control from an executing program to the supervisor program. Exceptions may be broadly divided into two categories: those caused by an instruction, including an unusual condition arising during its execution, and those caused by external events such as interrupts. When an IDT79R3000 detects an exception, the normal sequence of instruction is suspended; the processor exits User mode and is forced to the Kernel mode where it can respond to the abnormal or asynchronous event. This paper presents an overall view of the types of exceptions in the R3000 and the exception handling registers. It then describes one specific exception, namely interrupts, the latency associated with it and gives an example of code on how to handle an interrupt event.

EXCEPTION PROCESSING

The R3000's exception handling system efficiently handles machine exceptions, including Translation Lookaside Buffer (TLB) misses, arithmetic overflows, I/O interrupts, system calls, breakpoints, reset, and coprocessor unusable conditions. All of these events interrupt the normal execution flow. The R3000 aborts the instruction causing the exception and also aborts all those following in the exception pipeline which have already begun execution. The R3000 then performs a direct jump into a designated exception handler routine.

When an exception occurs, the R3000 loads the EPC (Exception Program Counter) with an appropriate restart location where execution may resume after the exception has been serviced. The restart location in the EPC is the address of the instruction causing the exception. If the exception occurred in a branch delay slot, the EPC contains the address of the branch instruction immediately preceding the delay slot.

EXCEPTION HANDLING REGISTERS

The system coprocessor (CP0) registers contain information pertinent to exception processing. Software can examine these registers during exception processing to determine such things as the cause of an exception, and the state of the CPU at the time of an exception. There are six registers handling exception processing (shown in Figure 1). These are the *Cause* register, the *EPC* register, the *Status* register, the *BadVAddr* register, the *Context* register, and the *Prld* register. A brief description of each follows.

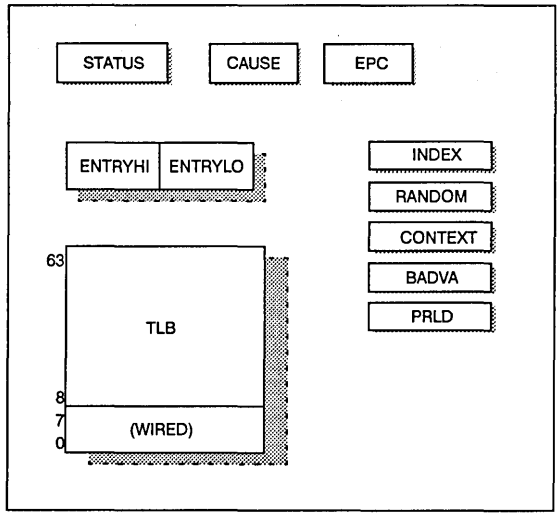
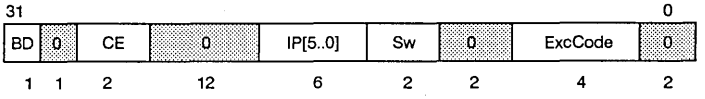


Figure 1. The Exception Handling Registers

The Cause Register:

The contents of this register describe the last exception. A 4-bit exception code indicates the cause. The remaining fields contain detailed information specific to certain exceptions. All bits in this register with the exception of the Sw bits are read-only. The Sw bits can be written into in order to set or reset software interrupts. See Figure 2.



BD : Branch delay.

CE : Coprocessor Error

IP : Interrupts Pending

Sw : Software Interrupts*

*: Read and Write. The rest are Read-only

ExcCode : Exception Code Field

0 : Reserved

Figure 2. The Cause Register

The EPC (Exception Program Counter) Register:

The 32-bit register contains the address where processing can resume after an exception has been serviced. This register contains the virtual address of the instruction that caused the exception. When the virtual address of the instruction resides in a branch delay slot, the EPC contains the virtual address of the instruction immediately preceding—which is the Branch or Jump instructions.

Bad VAddr Register:

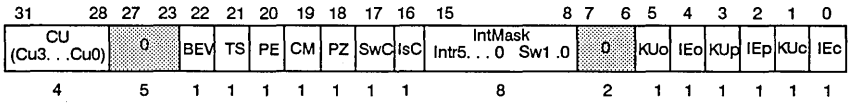
The Bad VAddr register saves the entire bad virtual address for any addressing exception.

Context Register:

The Context register duplicates some of the information in the BadVAddr register, but provides the information in a form that may be more useful for a software TLB exception handler.

The Status Register:

This register contains all the major status bits. Any exception puts the system in Kernel mode. All bits in the status register, with the exception of the TS (TLB shutdown) bit are readable and writeable; the TS bit is read-only. Figure 3 shows the functionality of the various bits in the status register.



CU : Coprocessor Usability

BEV : Bootstrap Exception vector

TS : TLB shutdown

PE : Parity Error

CM : Cache Miss

PZ : Parity Zero

SwC : Swap Caches

IsC : Isolate Cache

IntMask : Interrupt Mask

KUo : Kernel/User mode, old

IEo : Interrupt enable, old

KUp : Kernel/User mode, previous

IEp : Interrupt enable, previous

KUc : Kernel/User mode, current

IEc : Interrupt enable, current

O : Reserved

Figure 3. The Status Register

Processor Revision Identifier Register:

This 32-bit register contains information that identifies the implementation and revision level of the Processor and System Control Co-Processor.

EXCEPTION VECTOR LOCATIONS

The R3000 uses three different addresses for exception vectors:

- The RESET exception vector is at address *0xbfc00000*
- The UTLB Miss exception vector at address *0x8000000*
- The General exception vector for all other exceptions at address *0x80000080*

LATENCY FOR EXCEPTION PROCESSING

Different types of exceptions can occur in different stages of the pipeline. The exception handling routine itself occurs after a one cycle latency. The R3000 has a five stage pipeline that consists of instruction fetch, instruction decode, ALU operation, cache fetch, and the write-back stage. Table 1 shows in the last column the number of instructions in the pipeline that need to be flushed on an exception. Address error, for example, can have a maximum latency of four if it occurs on a memory operation cycle. This is because four instructions in the pipeline stage have to be flushed.

9

Error	Pipeline stage	Pipeline stages to be flushed
Illegal Instruction	Instruction Decode	2
Address Error	Memory Operation	4
Interrupts	Instruction Fetch	3
Overflow	ALU Operation	3
TLB Miss	Instruction Decode	2

Table 1. Latency in the R3000 on an exception

Figure 4 shows the pipeline stages in the R3000. The different stages are I:Instruction fetch; R:Read Decode; A:ALU operation; M:Memory operation; and W:Write-Back. When there is an exception on an instruction fetch cycle, the exception routine starts executing one clock cycle later as shown.

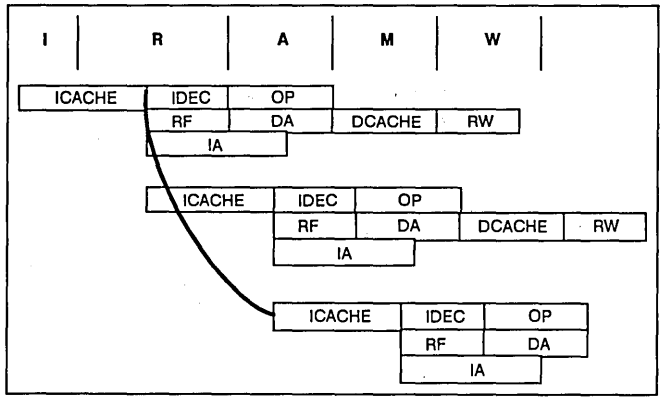


Figure 4. Pipeline Stages in the R3000

INTERRUPTS IN THE IDT79R3000

The R3000 processor has six general purpose hardware interrupts and two software generated interrupts. The hardware interrupts are sampled during phase 2 of all run and fixup clock cycles. This is shown in Figure 5. t_{DS} is the data setup time, t_{HLD} is the data hold time and t_{SMP} is the phase delay between the Clk2xSmp input and the Clk2xPhi input. These two clock inputs are part of the four phase clock inputs given to the processor and are useful for selecting the proper static RAM parts for interface considerations. The interrupts are level-sensitive. They continue to be sampled during phase 2 of the clock cycle after an interrupt exception has occurred. The interrupts are not latched within the processor when an interrupt exception occurs. Since the interrupts are not sampled during stall cycles, BusErr* can be asserted and used for exception processing. This is useful in cases where there is a need to abort from a stall mode.

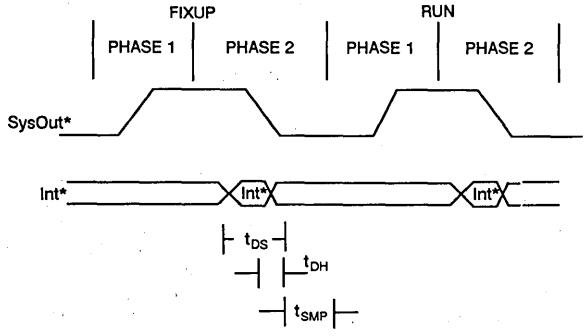


Figure 5. Interrupt Timing Diagram

Each of the eight interrupts can be individually masked by clearing the corresponding bit in the IntMask field of the Status Register. All eight of the interrupts can be masked at once by clearing the IEc bit in the Status Register.

INTERRUPT HANDLING

The R3000 branches to the general exception vector at address 0x80000080 for the exception. The R3000 sets the Int code in the Cause Register's ExcCode field. The IP field in the Cause register shows which of the six hardware interrupts are pending and the SW field in the Cause register shows which of the two software interrupts are pending. Multiple interrupts can be pending at one time.

When the interrupt occurs, the KUp, IEp, KUc and IEc bits of the Status register are saved in the KUo, IEo, KUp, IEp bit field in the Status register. The current kernel status bit KUc and the interrupt bit IEc are cleared. See Figure 6. This masks all the interrupts.

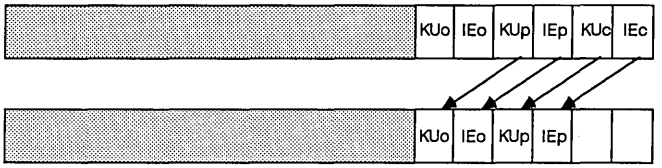


Figure 6. Kernel Status and Interrupt Status Are Saved on Interrupts

INTERRUPT SERVICING

In case of an hardware interrupt, the interrupt must be cleared by deasserting the interrupt line. This has to be done by alleviating the conditions that caused the interrupt. Software interrupts have to be cleared by setting the corresponding bits (SW1:0) in the Cause register to zero. A flow chart of a general exception routine handler is shown in Figure 7.

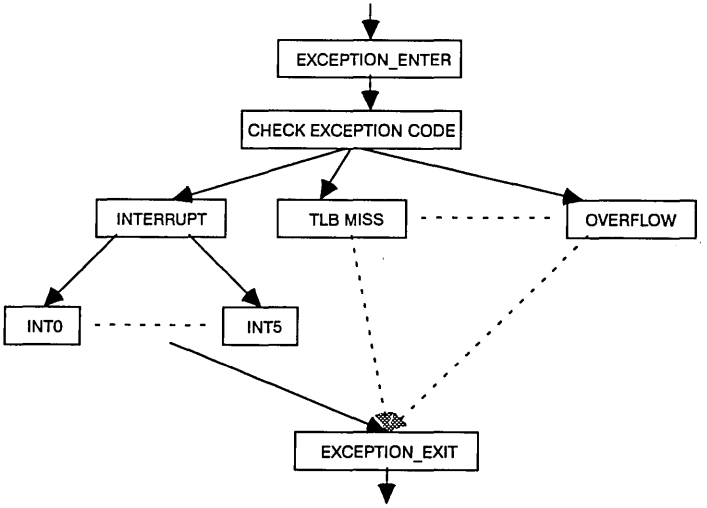


Figure 7. Flow Chart for Exception Handling

An example piece of code is given below in Figure 8. It illustrates a simple service routine that the processor branches to on detecting an interrupt. The actual interrupt handling code itself will depend on the user's application and, therefore, is not given. (an and sn are registers in the R3000.)

As soon as the branch to the address is taken on an interrupt, the exception program counter is saved. Line 2 indicates the reading of the cause register to determine the exception (in this case an interrupt). The status register is saved to be restored after processing the exception. A lookup table contains the addresses of all the different exception processing routines. A jump is taken to the appropriate exception routine.

```
1.    mfc0    a0,EPC          ;save exception PC
2.    mfc0    a3,C0_CAUSE     ;get CAUSE register
3.    mfc0    s0,C0_STATUS    ;save status register
```

```
4.          and     a1,a3,CAUSE_EXC_MASK   ;get entry to
                    index into look up table
5.          lw      a2,causevec(a1)        ;register a2 with address
                                           in look up table
6.          j       a2                     ;jump to service routine
            —                              (line 7)
7. intr:    and     a4,a0,INT_CAUSE        ;load interrupt level in
                                           register
8.          lw      t1,int_level(a4)       ; index into interrupt level
                                           'n' routine.
            —                              ;go to interrupt routine for
            —                              level 'n'
9.          j       t1                     ;return from interrupt
                                           routine
10.         mtc0    a0,EPC
11.         mtc0    s0,C0_STATUS           ;restore status register to
                                           previous value
12.         rfe                            ;restore status and
                                           interrupts prior to
                                           exception exit
```

Figure 8: Interrupt Service Routine Example

Figure 9 illustrates a simplistic block diagram of an R3000 board with the interrupt lines connected to a PAL device. The PAL logic is designed to affect the R3000 run-time behavior and it defines the state of the interrupt lines during Reset. Accordingly, the R3000 can be initialized to work as a big-endian or a little-endian processor, its block refill rate can be varied, etc.

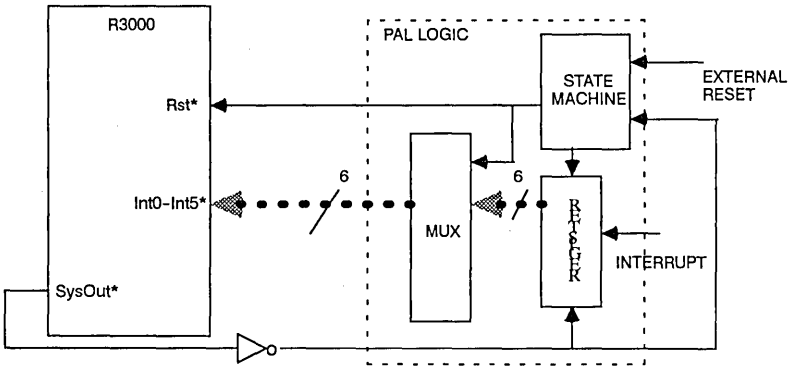


Figure 9: Block Diagram of Interrupt Controller on Reset

CONCLUSION

The IDT79R3000 provides both flexible and fast exception handling capability. Once an exception occurs, the first instruction of the exception handler routine is fetched on the very next clock cycle, providing minimal latency. Management of the processor and system state is left to the exception handling software, allowing the system designer to determine what must be done to respond to a given exception and thus minimizing the amount of processor overhead required to handle exceptions. Even the prioritization of the external interrupts is under software control, providing the system designer with maximum flexibility in the target system*.

*Note: Chapter 5 of the "MIPS RISC Architecture" Book, available from IDT, contains further detail on exception processing of the 79R3000.

Integrated Device Technology, Inc.

CACHE DESIGN CONSIDERATIONS USING THE IDT 79R3000

APPLICATION NOTE AN-27

By Satyanarayana Simha

INTRODUCTION

The reduced instruction set computer (RISC), the IDT79R3000, has allowed for simplicity in hardware and synergy between architecture and compilers. To further increase the throughput of a computer system, direct-mapped cache memory is implemented on systems using the R3000. The availability of a wide variety of high-speed static RAMs from IDT gives the designer the flexibility of selecting the proper part for his application. It is necessary, however, to know the critical timing parameters governing the design of a cache subsystem. This article is divided into three parts. The first part shows a general cache system with a description of the clock inputs. The second section details the equations used to calculate the critical parameters. It is followed by an example of an IDT7198 static RAM used as a cache RAM for the R3000.

CACHE DESIGN

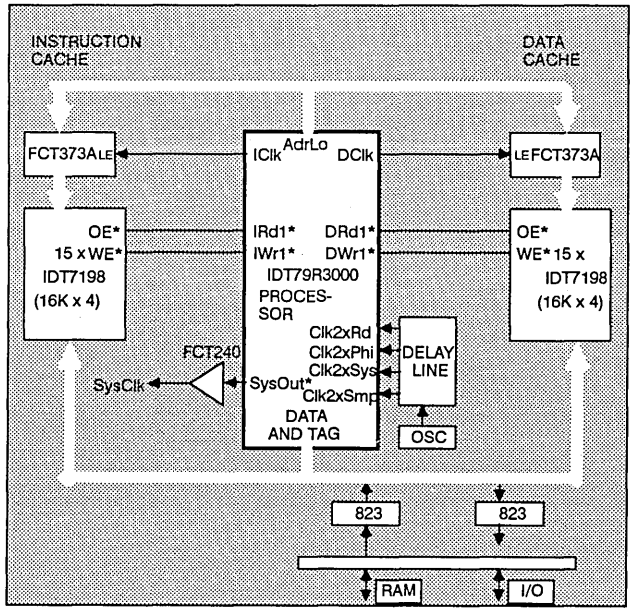


Figure 1. 64KB Instruction/Data Cache Configuration

A simplistic block diagram implementation of a 64KB separate instruction cache and data cache is shown in Figure 1. The design of a cache subsystem such as the one above depends on the four input clocks to the R3000 processor. These clock inputs are twice the frequency of the output clock i.e., SysOut. By adjusting the timings of these clocks, the designer can accommodate a wide variety of static RAMs by properly considering specific parameters such as set-up and hold times. The clocks themselves can be adjusted using tap settings on a delay line or by using delay logic. The clock inputs are described below.

1) Clk2xSyS: Determines the position of SysOut with respect to the data, tag, and address buses. It is positioned so that devices in the cache/bus interface clocked by SysOut meet the set-up and hold time requirements.
2) Clk2xSmp: Is used by the R3000 to capture external data onto data bus and control inputs.
3) Clk2xRd: Is used to delay the enable of data bus drivers.
4) Clk2xPhi: Is used to determine all R3000 outputs i.e., data, address, and tag buses.

Figure 2 shows the timing relationships between the four clocks. All the timing equations for cache design depend on the phase relationship between these clocks. T_{smp} is the Clk2xSmp to Clk2xPhi delay, T_{rd} is the Clk2xRd to Clk2xPhi delay, and T_{sys} is the Clk2xSys to Clk2xPhi delay.

9

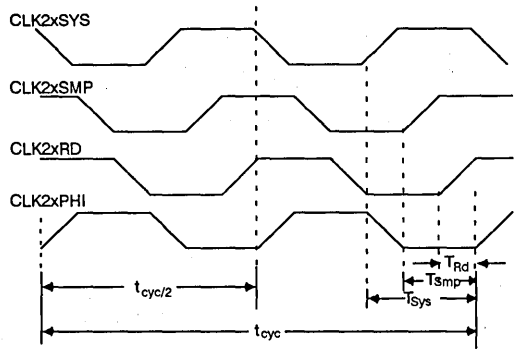


Figure 2. Timing Relationships Between the Four 2x Clock Inputs

In the cache implementation scheme, instruction references begin their reference during phase 2 and transfer data during the following phase 1. Data references begin during phase 1 and transfer data during phase 2. Thus, data and instruction references can take place in different phases of the same clock cycle. See Figures 3a and 3b. This is an important factor to consider in order to prevent contention between instruction and data caches.

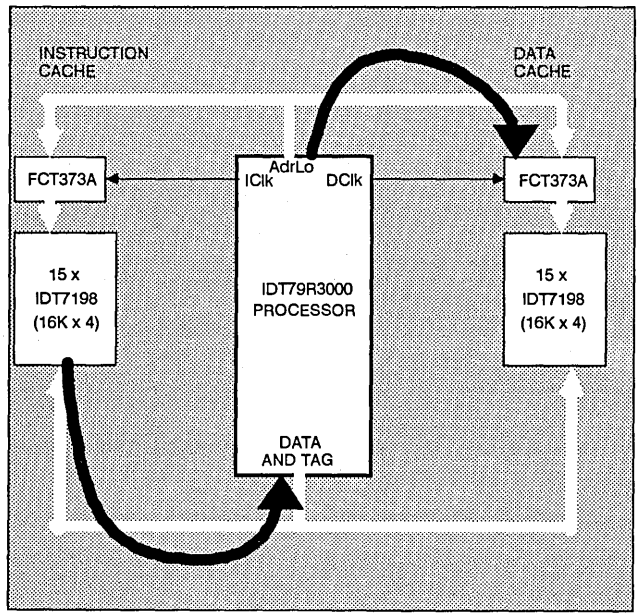


Figure 3a. Data and Instruction Caches During Phase 1

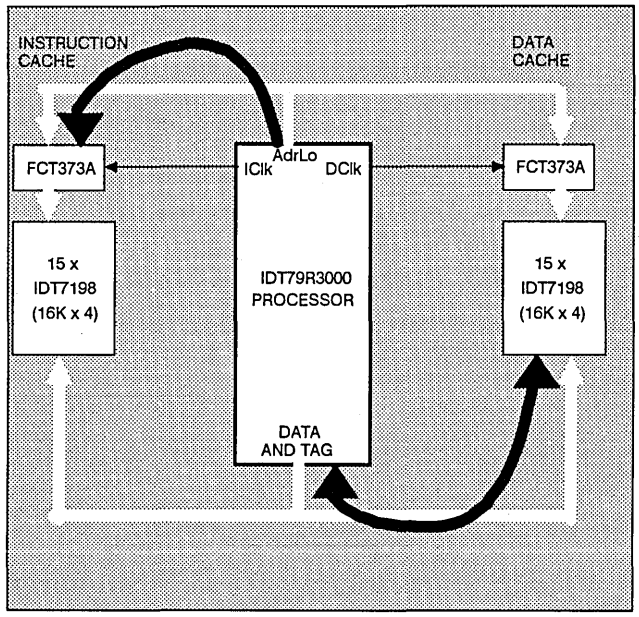


Figure 3b. Data and Instruction Caches During Phase 2

Specific factors such as access time, set-up time, hold time, enable and disable times, and the deration factor are key in choosing the proper static RAM and in setting the phase delays in the clocks. The next section discusses the timing equations needed for selecting a static RAM for cache design.

Equations Governing the Critical Parameters in RAM Selection:

Figure 4 shows the timing of the relevant signals for a 25MHz (40ns) R3000. The numbers represent the equations that are critical in determining the selection of the static RAMs. The timing is given for the worst case condition i.e., a STORE followed by a LOAD. An explanation of the equations is given below.

Internal Sample to Phi: This timing parameter requirement guarantees that the processor internal sample to Phi is met.

$t_{smp} \geq 5ns$ ---------- (1)

Address Access to Data Sample: This timing parameter requirement guarantees that the cache RAMs have sufficient access time. This calculation assumes that the address delay through the FCT373 is limited by its propagation delay.

$RAM^{d}_{AA} \leq t_{cyc} - t_{smp} - AdrLo^{d} - 373_{PD} - t_{os}$--- (2)

Cache Enable to Sample: This timing parameter requirement guarantees that the cache RAMs are enabled soon enough to meet the processor's input set-up specification.

$RAM_{OE}{}^{d} \leq T_{cyc/2} - t_{smp-rd} - Rd^{d} - t_{os}$------- (3)

Minimum Read Pulse Width: This timing parameter requirement guarantees that the read pulse generated by the processor is at least as long as the cache RAM output enable time.

$RAM^{d}_{OE} \leq t_{cyc/2} - t_{sys-rd}$ ---------- (4)

Read Write I-Cache Data Bus Contention: This timing parameter requirement guarantees that no contention will occur between the instruction cache and the processor on a store.

$RAM_{HZ} \leq t_{sys} - Rd^{d} + D_{en}$ ----- (5)

Processor Data Set-up to End of Write: This timing parameter requirement guarantees that the cache RAMs have adequate data set-up time when being written into by the processor.

$RAM_{SD} \leq t_{cyc/2} - t_{smp} - DVal^{d} + Wr^{d}$----- (6)

Data Hold from End of Write: This timing parameter requirement guarantees that the data hold from end of write specification of the cache RAMs is met when either the processor or the read buffer is writing to the RAMs.

$RAM_{HD} - RAM_{LZ} \geq t_{smp-rd}$ -------- (7)

Data Set-Up to SysClk: This timing parameter requirement guarantees that the set-up time into an external register is met on a processor store.

$SetUp_{sys} \leq t_{cyc/2} - t_{sys} - (DVal^{d} + - Sys^{d} - 240_{PD})$-- (8)

Data Hold from SysClk: This guarantees that the hold time specification of an external register is met on a processor store. The data holds on the bus until a subsequent read drives new data.

$Hold_{sys} \leq t_{sys-rd} - Sys^{d} - 240_{PD} + RAM_{LZ} + Rd^{d}$ (9)

Equations 1 to 9 are sufficient for the purpose of selecting the proper RAMs for use as cache memory. To illustrate the point further, an IDT RAM device, the IDT7198 (16K x 4), is chosen as an example.

*d: Deration due to additional load. 1ns per 25pF.

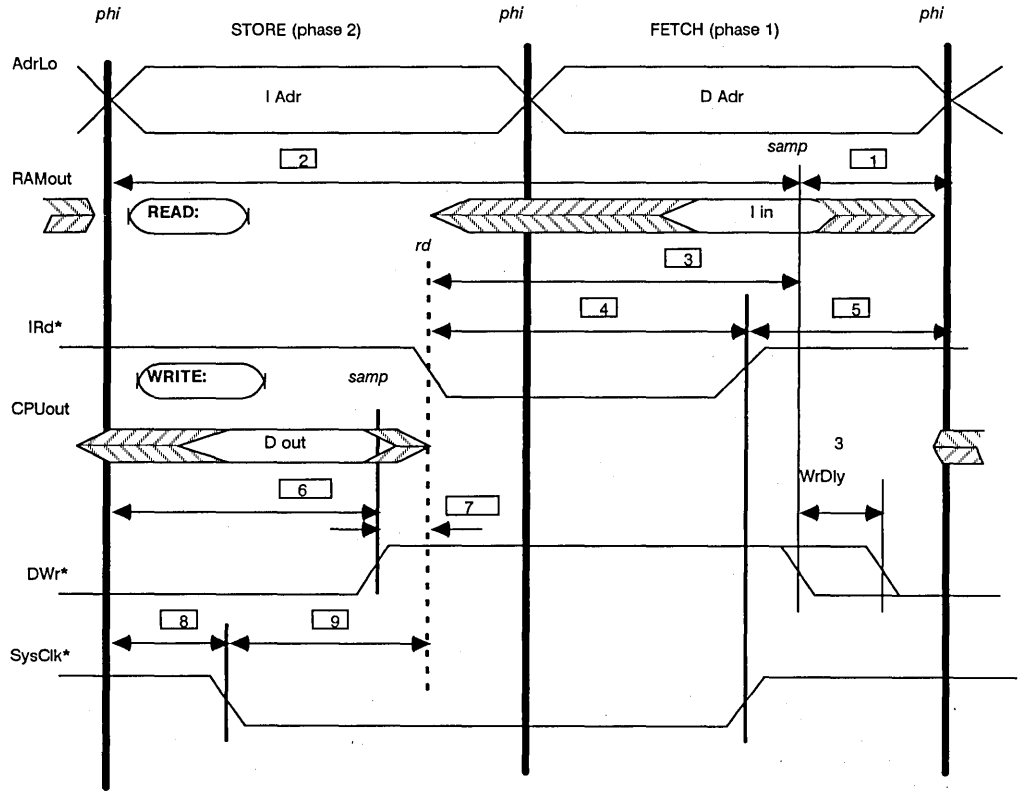


Figure 4. IDTR3000 40ns (25MHz) Cycle Timing

CALCULATION OF TIMING PARAMETERS

An example of a cache subsystem design using an IDT7198 i.e., a 16 K x 4 static RAM follows. All the numbers used in the calculations have been taken from the IDT Data Book[1] and the R3000 Interface Manual. The numbers are presented in Figure 6. The deration factor has been taken into account for DIPs. Surface mount would decrease the deration factor.

The following factors have been taken into account for calculating the deration factor[2].

1) The SSI logic and cache RAM propagation delays are derated by 1ns per 25pF of additional load.
2) Cache RAM input capacitance is 5pF.
3) Cache RAM output capacitance is 7pF.
4) Trace capacitance is estimated at 2pF per inch.
5) Data and trace tag buses are 6 inches.
6) Address buses are 2 x 5 inches.
7) SysOut loading is 50pF.
8) Test value of 30pF to be subtracted.

Deration Calculations:

Address Capacitance: 12 x 2pF = 24pF; (factors 4,5, and 6) 5 Devices = 5 x 10pF = 50pF; Test value = -30pF; Total capacitance = 45pF; At 1ns per 25pF, total deration of address bus = 2ns.

Data Bus Deration: Is approximately the same i.e., 2ns. Read control capacitance for IDT7198 will be about 10 inches of trace and 8 devices at 7pF each. Therefore, Read control deration = (76-30)pF/25pF/ns = 2ns.

In Figure 5, the circled numbers are the equations previously described. The number in parentheses is the allowable worst case timing. The adjacent number is the total time taken using the IDT7198. The numbers for the IDT7198 with the R3000 running at different frequencies and the FCT373A are shown in Table 1 and Table 2 respectively.

The first value for each parameter in Table 1 shows the maximum allowable worst case rating and the second value shows the timing using the IDT7198 RAM.

Parameter	Load (pF)	Symbol	R3000 Clock Frequencies					
			Min. (ns)	16MHz Max. (ns)		Min. (ns)	20MHz Max. (ns)	Min. (ns)	25MHz Max. (ns)
Address to Data Valid	30	t_{AA}		31			25		19
				29			25		19
Output enable to Data Valid	30	t_{DOE}		17			13		10
				15			13		10
Output Disable time		t_{HZ}		14			12	2	8
				12			10		7
Output Enable Time		t_{LZ}	2			2		2	
			5			5		5	
Address SetUp to End of Write		t_{AW}	43			36		27	
			20			20		13	
Data SetUp to End of Write		t_{DS}	14			13		11	
			13			13		8	
Write Pulse Width		t_{PWE}	55			47		37	
			20			20		13	
Data Hold from End of Write		t_{HD}	0			0		0	
			0			0		0	
Address Hold from End of Write		t_{HA}	0			0		0	
			0			0		0	

Table 1. Cache RAM Parameters. RAM Specifications vs. IDT7198 Specifications.

Parameter	Load (Units)	Symbol	Min. (ns)	Max. (ns)
FCT373A Propagation Delay	50	$t_{373\ PD}$		5.2
FCT373A Latch Enable Delay	50	$t_{373\ LE}$	2	8.5
FCT373A Latch Enable Hold	50	$t_{373\ Hld}$	1.8	
FCT240A Prop Delay	50	$t_{240\ PD}$	1.5	4.8

Table 2. Parameters for Latches and Buffers

Figure 6 shows a block diagram of tap settings on a delay line for the four clock input signals. By varying the phase delay between these signals, the designer can select the proper static RAMs for cache memories and the operating frequency of the R3000. Table 3 shows suggested tap settings on the delay line for the R3000 running at different frequencies.

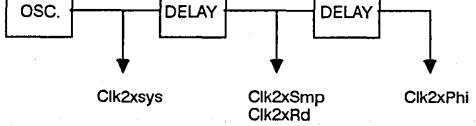


Figure 6. Tap Settings for the Clock Inputs

Parameter	16.67MHz	20MHz	25MHz
Clk2xSys	0	0	0
Clk2xRd	6	6	6
Clk2xSmp	6	6	6
Clk2xPhi	16	14	12

Table 3. Delay Line Setting Summarization

The designer can use a DDU-7F-20* chip for the delay line. The clock is the input to the device and the outputs at various points can be chosen with the appropriate phase delays.

CONCLUSION

The IDT R3000 RISC processor allows an efficient cache system to be implemented with standard architecture static RAMs. To design a cache subsystem, it is essential to know only the critical equations mentioned above and their relation to the four input clocks. The tap settings provide further control of the cache subsystem design for different operating frequencies of the R3000.

REFERENCES

1. IDT Data Book, pp 4-74 -- 4-83, pp 10-72 -- 10-75.
2. MIPS R3000 Processor Interface Manual, pp 105.

*d: deration due to additional load. 1 ns per 25 pF.

*Available from Data Delay Devices (201) 772-1106

9

Integrated Device Technology, Inc.

USING THE IDT79R3000 IN A MULTIPROCESSOR ORGANIZATION

APPLICATION NOTE AN-28

By Roy M. Johnson

INTRODUCTION

High performance systems, such as shared memory multiprocessor architectures, can be built using IDT79R3000 RISC processors. The IDT79R3000 incorporates special features that provide support for multiprocessor environments. This applications note discusses the features of shared memory multiprocessor architectures with local caches, examines the critical issue of cache coherency, and demonstrates how the features of the IDT79R3000 facilitate its use in a shared memory multiprocessor system.

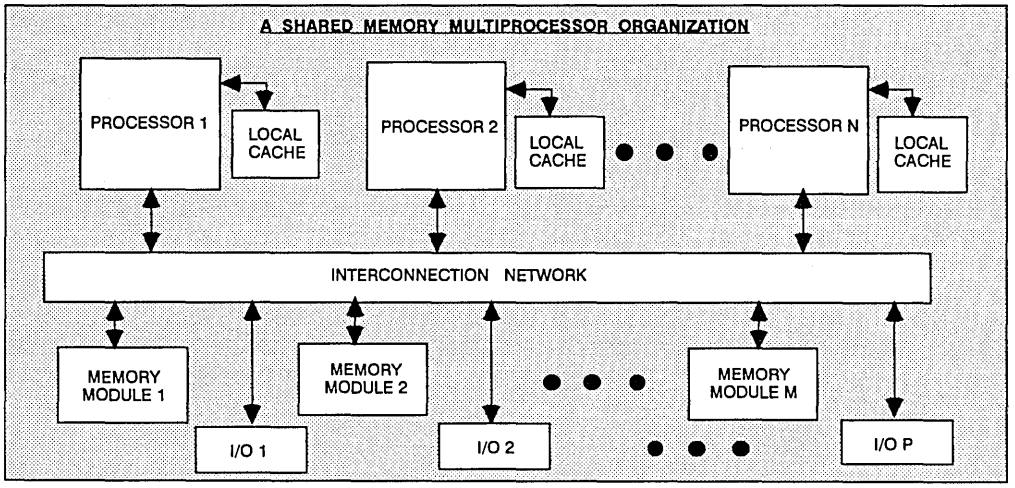


Figure 1. Block Diagram of a Shared Memory Multiprocessor System

SHARED MEMORY MULTIPROCESSOR SYSTEMS

A simplified block diagram of a shared memory multiprocessor with local caches is shown in Figure 1. This model of a multiprocessor system is defined to be tightly coupled and the N processors are connected to M memory modules and P I/O devices via an interconnection network. All the processors have a local cache memory, share the same global address space and communicate via shared memory. The interconnection network ensures complete connectivity between the processors and memory modules and can be implemented as a simple shared bus, multi-stage delta network or a more complex cross-bar switch. The global shared address space is assumed to be interleaved amongst the memory modules in order to minimize memory access conflicts. Note that the need for a interconnection network can be obviated by using a multi-port memory [1]. Examples of commercial machines employing a shared memory multiprocessor configuration using the R2000/3000 RISC processor include the Titan Graphics Supercomputer from Ardent Computers [2] and the 4D-MP Graphics Superworkstation from Silicon Graphics [3].

CACHE COHERENCY

The presence of local caches in a shared memory multiprocessor system introduces the issue of cache coherency that may result in data inconsistencies. This problem arises because several copies of the same data may exist in local caches of different processors at the same time. If one of the processors modifies (writes) the value of its copy of the data, then the other processors will have the stale or incorrect copy of the modified data in their local caches. This is a potential problem created by asynchronous parallel algorithms that do not have explicit synchronization. Data inconsistencies may also arise in multiprogrammed multiprocessor systems whereby a suspended process may migrate to another processor and the most recently updated data of the process might still be in the original processor's local cache. When the process is run on the new processor, there is a possibility that stale data is used if the local cache was not previously flushed. This assumes that the process did run previously on this processor. It is clear that if data consistency is to be ensured in a multiprocessor system, cache coherency must be maintained.

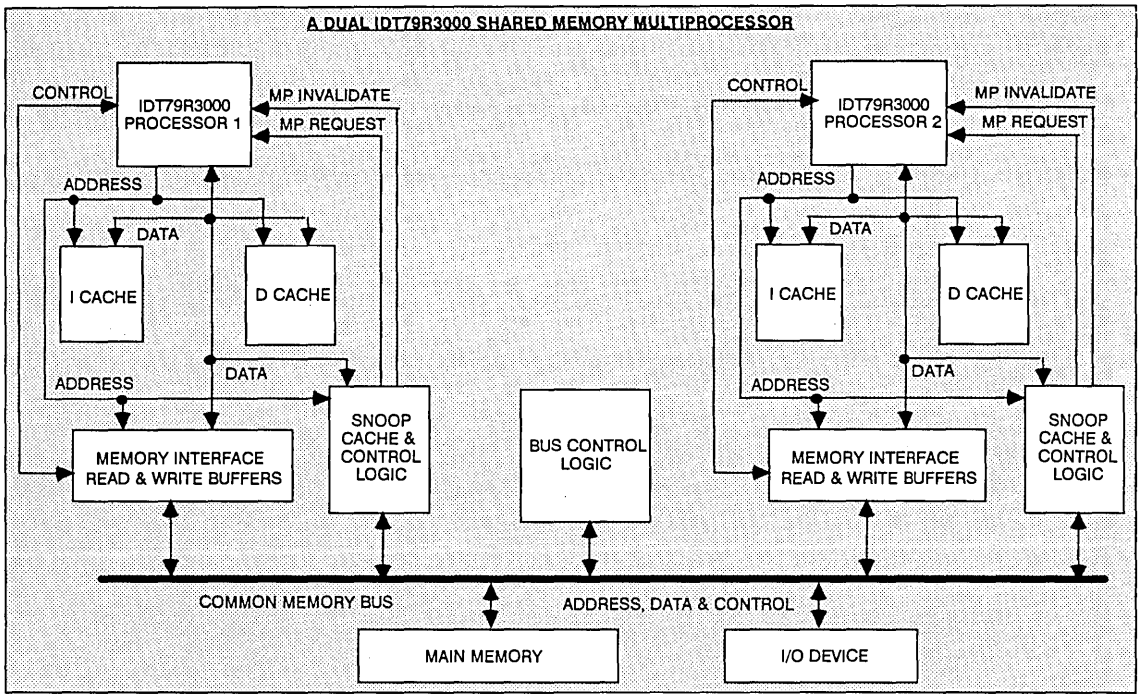


Figure 2. Block Diagram of a Dual IDT79R3000 Shared Memory Multiprocessor

A static approach to maintain cache coherency is to make all writeable data that is shared, non-cacheable. This method ensures data consistency, but at the price of decreased performance and with increased main memory conflicts. A dynamic approach to maintain cache coherency is to allow multiple copies of shared writeable data to exist and rely on a *cache coherence protocol* between the processors to ensure cache consistency. Several cache coherence protocols have been proposed and implemented using both hardware [4] and software support [5]. The type of protocol used depends primarily on interconnection network and the number of processors in the system.

A DUAL IDT79R3000 SHARED MEMORY MULTIPROCESSOR

A simplified block diagram of a dual IDT79R3000 shared memory multiprocessor is shown in Figure 2. A simple shared bus configuration was chosen for clarity. The two processors are connected to the main memory and an I/O device via a common bus. Access to the shared bus is arbitrated by the bus control logic. Each processor has an instruction and data cache and write-through cache update policy is assumed, i.e. all writes to the cache are also immediately transmitted directly to main memory. Note that a write-back cache update policy, (writes done only to the cache and main memory is updated when the cache line is replaced) would generate less memory traffic [10]. This is usually implemented when there are more than two processors in the system. Read and write buffers are included to provide a convenient asynchronous interface to the main memory. The snoop cache and control logic is used to implement a dynamic cache coherency check mechanism. For clarity, a very simple cache coherence protocol is chosen for the dual IDT79R3000 multiprocessor system and is described in detail below (more sophisticated and efficient schemes are described in [4], [5], [6], [7] & [8]).

9

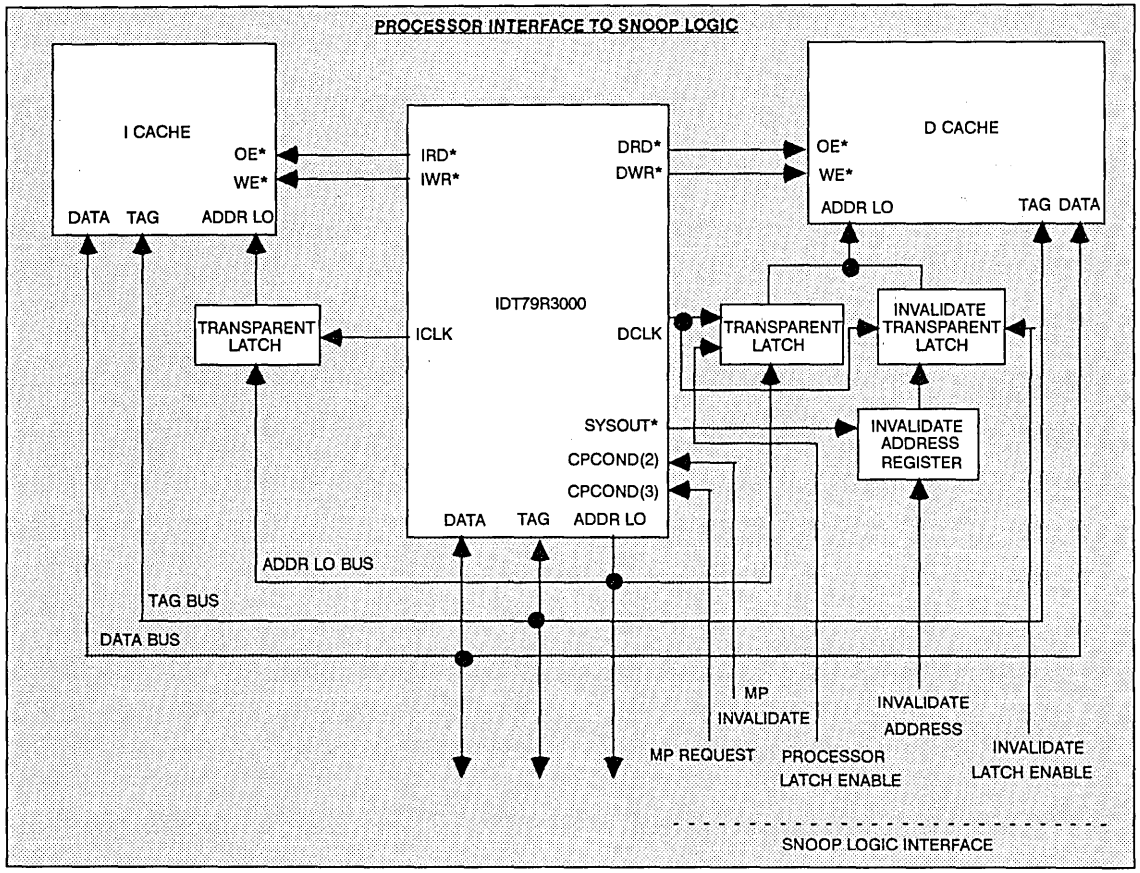


Figure 3. Processor Interface to the Snoop Control Logic

CACHE COHERENCE PROTOCOL

Each snoop cache maintains a directory of the current entries in the local data cache, (i. e. it contains the tags of all the current entries in the local data cache). Its primary function is to monitor the external memory bus for an address match. In addition, the snoop cache maintains state information for each data cache line. A cache (tag) line can be in one of three states: private, shared or invalid. Data that is exclusive to the processor is marked private, data that is common to the processors is marked shared and data that is inconsistent is marked invalid. The snoop cache is updated concurrently with the data cache. Whenever processor 1 modifies or writes a line that is marked shared in its local cache, its snoop control logic signals processor 2 that a write to a shared line has occurred. The snoop control logic of processor 2 then interrogates its snoop cache to determine whether a copy of the modified data is present in the local data cache. If a copy is present, it is invalidated using the *MP request* and *MP invalidate* signals as shown in the Figure 2 and the tag line in the snoop cache of processor 2 is marked invalid. The snoop control logic of processor 2 sends an acknowledge signal to processor 1 which then proceeds to complete its write operation to the shared location, i.e. writes into the data cache as well as into the write buffer. It must be noted that the data value in the write buffer must be retired to the main memory before the write operation can be completed. This prevents possible data inconsistencies that may arise by processor 2 trying to read that particular main memory location before it is updated. This cache coherence protocol is also known as *cross-interrogation*. Note that this protocol is applicable only to cache lines that are marked shared, while writes to cache lines marked private are performed at the processor speed. In the event of simultaneous writes to the same shared cache line by both the processors, only one of the processors will successfully acquire the external bus (determined by the bus arbitration logic) to issue a cross-interrogation signal to the other processor. The write operation of the processor that did not acquire the external bus will result in a write miss. Figure 3 shows a typical processor interface to the snoop cache and control logic in more detail, and is also described below.

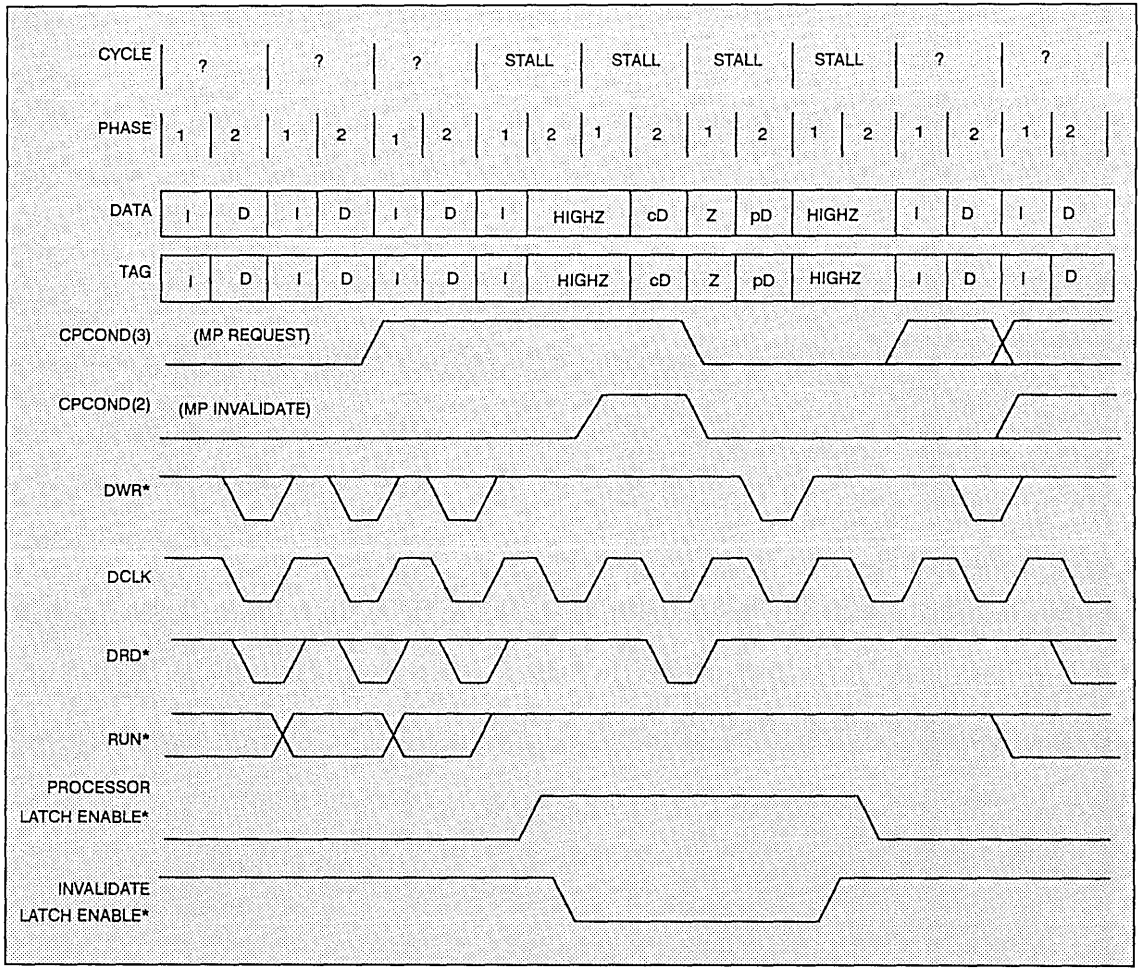


Figure 4. Cache Invalidation Timing Diagram

DYNAMIC CACHE COHERENCY CHECK MECHANISM

The signals at the snoop logic - processor interface include the *MP request*, *MP invalidate*, *processor latch enable*, *invalidate latch enable* and the *invalidate address* (address of the cache location to be invalidated). The snoop logic receives a cross- interrogation signal from the other processor when a write is performed to a shared cache line. It then searches its tags for an address match. If a match occurs, the address is captured in the invalidate address register which is clocked by *SysOut**, as shown in the Figure 3. The *CpCond(3)* input (*MP request* signal) of the IDT79R3000 is then asserted, causing the 79R3000 to enter into a MP stall. As there is no cache activity on the first cycle of an MP stall, the *processor latch enable* signal is deasserted and the *invalidate latch enable* is asserted in order to present the invalidate address to the data cache. After the first stall cycle, the CPU will issue DRd* pulses during every phase 2 and DClk (connected to the transparent latches) during every phase 1, this lasts until the end of the stall or until one cycle after the assertion of CpCond(2). This permits the snoop logic to read the data cache (Data and Tag values can be sampled by the falling edge of SysOut*) in order to determine whether an invalidation is to be performed. If the cache location is to be invalidated, the *MP invalidate* signal (connected to the CpCond(2) input of the 79R3000) is asserted. Invalidation occurs by the assertion of Dwr* during phase 2 of the stall cycle with an arbitrary invalid tag and arbitrary data value driven onto the Tag and Data buses. If CpCond(2) is deasserted while CpCond(3) is still asserted, the processor will return to issuing DRd* pulses to enable data cache reads. The cycle after CpCond(3) is deasserted contains no cache activity. This cycle is used to re-enable the processor's transparent latch and disable the invalidate transparent latch. A detailed timing diagram of a snapshot of the

9

cache invalidation process is shown in Figure 4. This is a modified version of the timing diagram shown in [11]. Note that Figure 4 shows the minimal timing required. CpCond(2) is asserted two cycles after CpCond(3) is asserted and before the first Drd*. This implies that the data location is invalidated irrespective of the value being read. The symbol "cD" denotes that the cache drives the data and tag buses when CpCond(3) is asserted. The symbol "pD" denotes that the processor drives the data and tag buses when CpCond(2) is asserted. The snoop control logic, at this stage, must mark the tag line in its snoop cache as invalid and send an acknowledge signal to the other processor. This indicates that the cache invalidation is complete. If desired, more sophisticated and efficient invalidation schemes, such as techniques for block invalidation, could be implemented.

SECONDARY CACHE SCHEME

The cache-main memory interface described above could be made more efficient by using a system of multi-level caches [9], [12], to provide additional memory bandwidth. For instance, a secondary cache that is four times the size of the first level or primary cache could be implemented. The secondary cache is a superset of the primary cache and also includes state information to maintain cache coherency. The cache update policy is typically write-through, from the primary to the secondary cache and write-back from the secondary cache to main memory. Since the primary cache is always a subset of the secondary cache, consistent data is guaranteed. This type of multi-level cache organization is implemented in the 4D-MP Graphics Superworkstation [3] made by Silicon Graphics.

CONCLUSION

Maintaining cache coherency is vital in shared memory multiprocessors. The implementation of the cache- main memory interface and the cache coherency protocol are critical issues. The IDT79R3000 RISC processor provides features that facilitate the implementation of cache coherence check mechanisms with minimum hardware and is well suited to be used in a shared memory multiprocessor environment.

REFERENCES

[1] K. Hwang & F. A. Briggs, *Computer Architecture and Parallel Processing*, McGraw-Hill, 1984, pp 459 - 525.

[2] T. Diede et al, "The Titan Graphics Supercomputer Architecture", *IEEE Computer*, Sept. 1988, pp 13 - 30.

[3] F. Baskett, T. Jermoluk & D. Solomon, "The 4D-MP Graphics Superworkstation: Computing + Graphics = 40 MIPS + 40 MFLOPS and 100,000 Lighted Polygons per second", Digest of Papers, COMPCON, Spring 1988, 33'rd IEEE Computer Society Int. Conference, pp 468 - 471.

[4] J. Archibald & J. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model", *ACM Transactions on Computer Systems*, Vol. 4, No. 4, Nov. 1986, pp 273 - 298.

[5] A. J. Smith, " CPU Cache Consistency with Software Support and Using "One Time Identifiers" ", Technical Report 86/290, EECS Department, University of California, Berkeley, California 94720.

[6] M. Papamarcos & J. Patel, "A Low-Overhead Coherence Solution for Multiprocessors with Private Cache Memories", Proc. 11'th Annual Int. Symp. on Comp. Arch., June 1984, pp 348 - 354.

[7] L. Rudolph & Z. Segall, "Dynamic Decentralized Cache Schemes for MIMD Parallel Architectures", Proc. 11'th Annual Int. Symp. on Comp. Arch., June 1984, pp 340 - 347.

[8] P. Sweazey & A. J. Smith, "A Class of Compatible Cache Consistency Protocols and their Support by the IEEE Futurebus", Proc. 13'th Annual Int. Symp. on Comp. Arch., June 1986, pp 414 - 423.

[9] A. J. Smith, "Design of CPU Cache Memories", Proceedings, IEEE TENCON, Korea, Aug. 1987, pp 1 -10.

[10] A. J. Smith, "Cache Memories", ACM Computing Surveys, Vol. 14, No. 3, Sept 1982, pp 474 - 530.

[11] Multi-processor Interface, MIPS R3000 Processor Interface, MIPS Computer Systems, May 23 1988, pp 59-61.

[12] S. Przybylski, M. Horowitz & J. Hennessey, "Performance Tradeoffs in Cache Design", Proc., 15th Annual Int. Symp. on Comp. Arch., June 1988, pp 1 - 18.

Integrated Device Technology, Inc.

THE EFFECT OF BRANCH, LOAD AND STORE LATENCY ON RISC PROCESSOR PERFORMANCE

APPLICATION NOTE AN–33

By Roy M. Johnson

INTRODUCTION

One of the ways RISC processors attempt to achieve single-cycle execution is by pipelining instruction execution. Instruction flow through the pipeline can be impeded, however, due to the latency effects of instructions such as loads, stores and branches. As a consequence of these latency effects, single-cycle instruction execution cannot be maintained without a loss of pipeline efficiency. This article uses a simple probabilistic model to measure the effect of load, store and branch instructions on pipeline performance in the Am29000, MC88100, SPARC and the R3000 RISC processors. The analysis shows that the pipeline architecture of the R3000 processor is able to minimize the latency effects of loads and branches to a greater extent than the other RISC processors and thereby maintain higher pipeline efficiency.

INSTRUCTION PIPELINE

An instruction pipeline can potentially increase the throughput of a processor by overlapping the execution of several different instructions. In other words, at any given time in an instruction pipeline, several instructions can be in different stages of execution. Figure 1 shows the stages of a typical instruction pipeline of a RISC processor with a load/store architecture (only load and store instructions access memory and all other operations are performed on registers). After the initial start up latency to fill the pipeline, an instruction theoretically can complete execution on every clock cycle as shown in Figure 2.

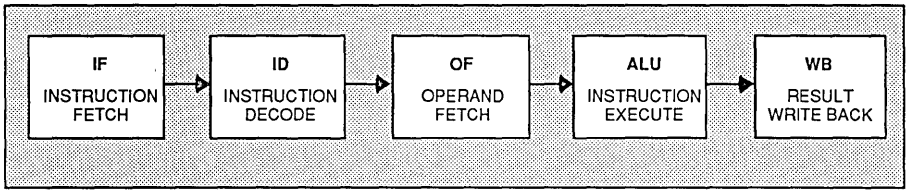


Figure 1. Stages of a Typical R3000 Instruction Pipeline

However, because of data dependencies between instructions in the pipeline and latency effects of instructions such as loads and branches, single-cycle instruction execution cannot be maintained. For instance, in the case of conditional branch instructions, if the branch condition is evaluated in the ALU stage of the pipeline (as shown in Figure 1) at a time t, and if the branch condition is true, the target instruction can only be fetched at time t + 1. The instructions fetched at times t – 3, t – 2 and t –1 (assuming sequential instruction fetch continues after the branch instruction is fetched) will have to be discarded or flushed from the pipeline. Thus, there are 3 cycles in which no useful work is done. Similarly, in the case of a load instruction, because of finite memory access time, the operand of the load instruction will not be available for the next instruction in the pipeline (typically the instruction after the load uses the data of the load instruction). Therefore, the pipeline will have to be stalled while the memory request is satisfied. Store instructions can similarly stall the pipeline if both instructions and data use the same single port memory. To prevent memory access conflicts, the instruction after a store cannot be fetched until the store is completed. The examples described above illustrate the loss in pipeline efficiency due to the latencies of branch and memory reference instructions.

Techniques to improve pipeline efficiency by reducing the branch penalty include the use of branch target buffers [1], dynamic branch prediction strategies and delayed branching schemes [2]. Latency effects of memory reference instructions can be minimized by the use of a Harvard architecture with a split cache memory for instruction and data [3] and instruction prefetching schemes.

The effect of load, store and branch instructions on pipeline performance can be approximated using a simple probabilistic model. This mathematical model is an extension of the one proposed by D. J. Lilja [4] for measuring the effect of branch penalty in pipelined processors. The model is described below and is used to compare the pipeline performance of the Am29000, MC88100, SPARC and the R3000 RISC processors.

CYCLE	PIPELINE STAGE					
	IF	ID	OF	ALU	WB	
1	i					(start up latency)
2	i + 1	i				"
3	i + 2	i + 1	i			"
4	i + 3	i + 2	i + 1	i		"
5	i + 4	i + 3	i + 2	i + 1	i	(instruction i completes)
6	i + 5	i + 4	i + 3	i + 2	i + 1	(instruction i + 1 completes)
7	•	•	•	•	•	etc.

Figure 2. Instruction Flow Through the Pipeline

9

PIPELINE PERFORMANCE MODEL

Consider a pipelined RISC processor in which all instructions except memory reference instructions and branches can be executed on every clock cycle, i.e. assume single cycle execution for all instructions except loads, stores and branches. Then the average number of cycles per instruction (CPI_{ave}) is given by:

$$CPI_{ave} = P_b(1 + b) + P_m(1 + m) + (1 - P_b - P_m)(1) \quad (1)$$

CPI_{ave}= average number of cycles per instruction
P_b = probability that an instruction is a branch
b = branch penalty (number of cycles wasted in the pipeline if an instruction is a branch)
P_m = probability that an instruction is a memory reference (load or store)
m = memory reference penalty (number of cycles wasted in the pipeline if an instruction is a memory reference, i.e. the memory reference latency)

Pipeline efficiency can then be computed as the ratio of the minimum number of cycles per instruction (maximum performance case, which is 1 cycle for this model) to the average number of instructions per cycle:

$$Pipeline_{eff} = \frac{1}{CPI_{ave}} \quad (2)$$

where $Pipeline_{eff}$ = the pipeline efficiency

Note that if b = 0 and m = 0, i.e. there are no branch or memory reference penalties, then

CPI_{ave} = 1 from Equation (1)

and $Pipeline_{eff}$ = 1 from Equation (2)

In other words, when the average number of cycles per instruction is 1, then the pipeline operates at maximum efficiency.

The key assumptions of the pipeline model represented by Equation 1 are:

1) The execution of all branch and memory reference instructions will result in a pipeline penalty. This may not be true in all RISC architectures especially with reference to conditional branch instructions where the pipeline penalty will depend on whether or not the branch is taken. Similarly, the memory reference penalty will depend on whether the instruction is a load or a store.

2) The model also assumes that a new instruction can be fetched on every clock cycle, i.e. a 100% instruction cache hit rate which is the ideal case.

Equation 1 should be modified to accurately model a target architecture. The RISC architectures in this comparison (the Am29000, SPARC, MC88100 and R3000) use a delayed branching scheme. A processor with a branch delay of b cycles will always execute b instructions after a branch instruction, whether the branch is taken or not. The compiler (for a processor using delayed branches) determines the instruction dependencies and reorganizes the instruction stream to fill the branch delay slots with useful instructions. In many cases some or all of the b delay slots after the branch must be filled with no-ops. Gross and Hennessy [5] developed an algorithm for optimizing delayed branches in the MIPS project and have shown that the first branch delay slot can be filled with a useful instruction more than half the time while subsequent delay slots are increasingly harder to fill. Equation 1 can be modified for a RISC architecture with a delayed branching scheme with b branch delay slots as follows:

$$CPI_{ave} = P_b(1 + b_{nop}) + P_m(1 + m) + (1 - P_b - P_m)(1) \quad (3)$$

where, b = number of delay slots after a branch instruction the effective number of cycles/branch = $1 + bP_{nop}$ and the effective number of cycles/memory reference = 1 + m
if f_i is the probability that a delay slot i is filled with a useful instruction, then

$$P_{nop} = 1 - \frac{f_1 + f_2 + \ldots\ldots + f_b}{b} \quad (4)$$

Thus, P_{nop} shows the fraction of delay slots that do no useful work and thereby add to the branch penalty.
Equation 3 can be further reduced to:

$$CPI_{ave} = 1 + bP_bP_{nop} + mP_m$$

In other words the average number of cycles per instruction is 1 plus a fraction due to the branch penalty plus a fraction due to the memory reference penalty. The pipeline model represented by Equation 3 will be used in the architectural comparisons made in the subsequent section.

INSTRUCTION MIX

The format of the instruction mix used in the performance comparisons is shown in Table 1. This instruction mix is typically representative of non numeric code (no floating point operations included) and is obtained from a suite of integer benchmarks performed by MIPS Computer Systems. It is important to note that even though an instruction mix is program dependent, a fairly significant portion of the total number of instructions usually consists of load, store and branch instructions. In the analysis it is assumed that conditional branch instructions are taken 65% of the time [4] and the instruction immediately following a load always uses the data of the load instruction.

INSTRUCTION TYPE	RELATIVE FREQUENCY
ALU operations	.55
Branch instructions	.15
Load Instructions	.20
Store Instructions	.10
TOTAL	1.0

Table 1. Instruction Mix Format

PERFORMANCE COMPARISONS

This section uses the pipeline model represented by Equation 3 and the instruction mix described in Table 1, to compare the the pipeline efficiencies of the Am29000, SPARC, MC88100 and the R3000 RISC architectures. It is assumed that the branch delay slot can be filled with a useful instruction 50% of the time in all cases [5].

Am29000

The Am29000 RISC processor has a 4 stage instruction pipeline, provides an interface to separate instruction and data caches and also includes an on-chip branch target cache [7]. The processor also includes a 4 deep instruction pre-fetch buffer. All branch instructions execute with a single branch-delay slot. If the branch is

taken and the target address is found in the branch target cache, there is no extra branch penalty (except for the single delay slot). However, in case of a branch target cache miss, there is a penalty of at least 2 cycles and the probability of a branch target buffer miss will be assumed to be .25 [4]. Load instructions execute with a latency of at least 1 cycle and a store instruction will cause a pipeline hold mode of at least one cycle if overlapped with a load. Equation 3 can be modified to to model the Am29000 and reduces to:

$$CPI_{ave} = 1 + bP_bP_{nop} + P_bP_t P_m c + 1P_i + SP_s \quad (6)$$

CPI_{ave} = average number of cycles per instruction
P_b = probability that an instruction is a branch (.15)
b = number of branch delay slots (1)
P_{nop} = probability that an instruction in a branch delay slot is a no-op (.5)
P_t = probability that branch is taken (.65)
P_m = probability of a miss in the branch target buffer (.25)
c = number of cycles wasted due to a branch target buffer miss (2)
P_l = probability that an instruction is a load (.2)
P_s = probability that an instruction is a store (.1)
l = load latency (1)
s = store latency (1)

effective cycles/load = 1 + l = 2
effective cycles/store = 1 + s = 2
effective cycles/branch = $1 + bP_{nop} + P_t\ P_{tm}\ c = 1.825$

The numbers in parentheses are the actual values assigned to each of the factors. Using these values in Equation 6, the average number of cycles per instruction is computed to be:

$$CPI_{ave} = 1.424 \text{ cycles}$$

and the pipeline efficiency is computed to be

$$Pipeline_{eff} = 1 / 1.424 = .702$$

from Equation 2

Therefore, for the instruction mix described in Table 1, the pipeline operates at an efficiency of 70.2%.

SPARC

The SPARC processor has a 4 stage instruction pipeline, includes a large register file (136 general purpose registers) [8] and uses a register windowing scheme for parameter passing. It can be argued that in the case of a RISC architecture with a register windowing scheme, there will be relatively fewer loads and stores for a given instruction mix. The code generated, however, depends on the quality of the compiler. Morrison and Walker [13] have shown that for a given instruction mix the compiler for the SPARC architecture (the only architecture in this analysis that actually uses a register windowing scheme) reduced the the percentage of loads and stores by 3-9% compared to the MIPS compiler. The MIPS optimizing compiler, however, generated fewer total instructions and sometimes up to 1/3 fewer instructions compared to the SPARC compiler. The MIPS compiler therefore, generated fewer total number of loads and stores. The instruction mix in Table 1 is used for comparing the performance of the SPARC with the R3000. The SPARC processor has a single address and data bus and uses a single cache for both instructions and data. The architecture includes 2 deep instruction pre-fetch buffer and uses a value in the condition code register for conditional branching. A delayed branching scheme is implemented with a branch delay slot of 1 cycle. The target instruction of a branch is always fetched regardless of whether the branch is taken or not. In the case of a conditional branch that fails, there is an additional penalty to flush the pipeline and fetch the correct instruction sequence [6]. Loads take 2 cycles; the address of the operand is computed in the ALU stage (stage 3) of the pipeline and the cache access is made in stage 4. A new instruction cannot be fetched during stage 4 of a load instruction as both instruction and data share the same memory and therefore a load effectively takes 2 cycles.For similar reasons, store instructions take 3 cycles . If it is assumed that there is a branch penalty of at least 2 cycles for a conditional branch that fails (i.e. to flush the first two stages of the pipeline), then equation 3 then reduces to:

$$CPI_{ave} = 1 + P_b\ bP_{nop} + (1 - P_t)\ 2) + 1P_l + sP_s \quad (7)$$

P_b = probability that an instruction is a branch (.15)
b = number of branch delay slots (1)
P_{nop} = probability that an instruction in a branch delay slot is a no-op (.5)
P_t = probability that branch is taken (.65)
P_l = probability that an instruction is a load (.2)
P_s = probability that an instruction is a store (.1)
l = load latency (1)
s = store latency (2)

effective cycles/load = 1 + l = 2
effective cycles/store = 1 + s = 3
effective cycles/branch = $1 + bP_{nop} + (1 - P_t\ )2_t = 2.2$

Using these values in Equation 7, the average number of cycles per instruction is computed to be:

$$CPI_{ave} = 1.58 \text{ cycles}$$

and the pipeline efficiency is computed to be

$$Pipeline_{eff} = 1 / 1.58 = .633$$

from Equation 2

Therefore, for the instruction mix described in Table 1, the pipeline operates at an efficiency of 63.3%.

MC88100

The Motorola MC88100 RISC processor includes 32 general purpose registers and uses a two-port, non multiplexed memory access scheme (Harvard architecture). The processor contains a 2 stage instruction unit pipeline that fetches and supplies instructions to the integer or floating-point unit [9]. Data memory accesses are pipelined and controlled by the 3 stage data unit. Because the integer unit consists of a single stage, there are effectively 6 pipeline stages for an integer ALU operation performed on operands in memory. A delayed branching scheme is employed with a delay slot of 1 cycle. A separate adder for address calculations is also included so that branch target addresses are computed in parallel with instruction decoding. The branch penalty therefore is only 1 cycle (the single delay slot). The load latency is 2 cycles because the memory management unit is off-chip and additional time is needed for address translation; stores on the other hand have no latency and execute in a single cycle. Because loads and stores are pipelined, there are possible memory conflicts that will add to the memory reference penalty. It is assumed that the compiler can reorganize the instruction stream so that the first load delay slot (first load latency cycle) can be filled with a useful instruction 70% of the time and the second delay slot can be successfully filled 30%

of the time. Therefore, the probability that the instructions in the load delay slots are no-op's can be computed using Equation 4. Equation 3 then reduces to:

$$CPI_{ave} = 1 + bP_b P_{nopb} + 1P_l P_{nopl} \quad (8)$$

P_b = probability that an instruction is a branch (.15)
b = number of branch delay slots (1)
P_{nopb} = probability that an instruction in a branch delay slot is a no-op (.5)
P_l = probability that an instruction is a load (.2)
l = number of load delay slots (2)
P_{nopl} = probability that the instructions in the load delay slots are no-ops (.5)

effective cycles/load = $1 + lP_{nopl}$ = 2
effective cycles/store = 1
effective cycles/branch = $1 + bP_{nopb}$ = 1.5

Using these values in Equation 8, the average number of cycles per instruction is computed to be:

$$CPI_{ave} = 1.275 \text{ cycles}$$

and the pipeline efficiency is computed to be

$$Pipeline_{eff} = 1 / 1.275 = .784$$

from Equation 2

Therefore for the instruction mix described in Table 1, the pipeline operates at an efficiency of 78.4%.

R3000

The R3000 RISC processor [10] has a 5-stage instruction pipeline and includes 32 general purpose registers. The processor has both an address and a data bus which are cycled at twice the processors clock frequency. This enables the processor to have separate instruction and data caches. The instruction cache is accessed during one phase of the processor cycle, while the data cache is accessed on the other phase [11]. All branch instructions execute with a delay slot of 1 cycle. The processor also includes a separate adder for address computations which enables the target address of a branch instruction to compute in parallel with the instruction decode phase of the pipeline [12]. The processor employs a fast compare scheme for conditional branches (includes a separate comparator for equality, inequality and any relation with zero tests). Branch conditions, therefore, can be evaluated early in the pipeline and the ALU stage is not required for most conditional branches [2]. Thus, the branch penalty is 1 cycle (branch delay slot) that the compiler successfully fills 50% of the time. Load instructions have a latency of 1 cycle (the address of the load is computed in stage 3, and the memory access is completed by the end of stage 4) and it is assumed that the compiler fills the load delay slot with a useful instruction 70% of the time. Store instructions have no latency and execute in a single cycle. Using these parameters Equation 3 reduces to:

$$CPI_{ave} = 1 + bP_b P_{nopb} + 1P_l P_{nopl} \quad (9)$$

P_b = probability that an instruction is a branch (.15)
b = number of branch delay slots (1)
P_{nopb} = probability that an instruction in a branch delay slot is a no-op (.5)
P_l = probability that an instruction is a load (.2)
l = number of load delay slots (1)
P_{nopl} = probability that the instruction in the load delay slots is a no-op (.3)

effective cycles/load = $1 + lP_{nopl}$ = 1.3
effective cycles/store = 1
effective cycles/branch = $1 + bP_{nopb}$ = 1.5

Using these values in Equation 9, the average number of cycles per instruction is computed to be:

$$CPI_{ave} = 1.135 \text{ cycles}$$

and the pipeline efficiency is computed to be

$$Pipeline_{eff} = 1 / 1.135 = .881$$

from Equation 2

Therefore, for the instruction mix described in Table 1, the pipeline operates at an efficiency of 88.1%.

CONCLUSION

The performance comparisons made above are summarized in Table 2 shown below. The effective number of cycles for branch and memory operations are also included. It is important to note that the CPI_{ave} and the $Pipeline_{eff}$ results shown in Table 2 pertain to the instruction mix described in Table 1. Any instruction mix that has a greater frequency of loads, stores or branches will increase the average number of instructions per cycle and thereby decrease the pipeline efficiency. The results show the detrimental effect of load, store and branch latencies on pipeline efficiency. The analysis shows that the R3000 maintains the highest pipeline efficiency for the given instruction mix. This is because the pipeline architecture of the R3000 is optimally designed to minimize the latency effects of loads and branches. The branch latency in the R3000 is minimized by using a fast single cycle compare and branch and having the branch target address computed by the second stage of the pipeline. The load latency is minimized by having the cache control and tag compare logic on chip and accessing the instruction and data caches on alternate phases of the clock.

PROCESSOR	Am29000	SPARC	MC88100	R3000
Effective cycles/load	2	2	2	1.3
Effective cycles/store	2	3	1	1
Effective cycles/branch	1.825	2.2	1.5	1.5
CPI_{ave}	1.424	1.58	1.275	1.135
$Pipeline_{eff}$	70.2%	63.3%	78.4%	88.1%

Table 2. Performance Comparison Summary

The effect of instruction cache misses has not been taken into account in this analysis and it is critical that the cache miss penalty is minimized. In conclusion, the analysis shows that the latency effects of memory reference and branch instructions are critical to pipeline performance. The R3000 processor is able to minimize these latencies to a greater extent than the Am29000, SPARC and MC88100 processors and thus obtain better pipeline efficiency.

REFERENCES

[1] J. K. F. Lee and A. J. Smith, "Branch Prediction Strategies and Branch Target Buffer Design", *IEEE Computer*, Jan. 1984, pp 6 – 22.

[2] S. McFarling and J. Hennessy, "Reducing the Cost of Branches", *Proc. 13th Annual Symp. on Computer Architecture*, 1986, pp 396 – 403.

[3] A. J. Smith, "Cache Memories", *ACM Computing Surveys*, Vol. 14, No. 3, Sept. 1982, pp 474 – 530

[4] D. J. Lilja, "Reducing the Branch Penalty in Pipelined Processors", *IEEE Computer*, July 1988, pp 47 – 55.

[5] T. R. Gross and J. L. Hennessy, "Optimizing Delayed Branches", *Proc. 15th annual Workshop on Microprogramming*, Oct. 1982, pp 114–120.

[6] F. Baskett, "SGI's 4D MIPS processor versus the Sun – 4 SPARC processor", correspondence via electronic mail on the UNIX network to the comp.arch newsgroup, Dec 10, 1987.

[7] Am29000 Streamlined Instruction Processor User Manual, Advanced Micro Devices Inc., 1988, pp 4–1 to 4–16.

[8] SPARC MB86901 High Performance 32-bit RISC Processor, Product Description, Fujitsu Microelectronics Inc., Sept. 1988, pp 2 – 30.

[9] MC88100 RISC Microprocessor User's Manual, Motorola Inc., 1988.

[10] G. Kane, *MIPS RISC Architecture*, Prentice-Hall Inc., 1987, pp 1–7 to 2–15.

[11] T. Riordan and P. Ries, *MIPS R3000 Processor Interface*, MIPS Computer Systems Inc., Oct 27 1988, pp 4 – 7.

[12] T. Riordan et al, "The MIPS M2000 System", *Proc. International Conference on Computer Design (ICCD)*, Oct. 1988, pp 366 – 369.

[13] S. Morrison and N. Walker, "Register Windows Vs. General Registers: A Comparison of Memory Access Patterns", Technical Note, University of California, Berkeley, California, May 13, 1988.

Integrated Device Technology, Inc.

DESIGN TECHNIQUES FOR LOWERING POWER CONSUMPTION IN RISC DESIGNS

APPLICATION NOTE AN-55

by Barry Seldner, FAE

Most devices have two variables that relate to power consumption — static or standby power and dynamic or operational power. Dynamic power is consumed at a rate that is almost linear with speed. A CMOS rule is that the faster a device toggles, the more power is consumed. This exemplifies the typical CMOS characteristic where power is consumed during transition time. Operation, therefore, results in power consumption added to the static-power base line.

Applications using RISC processors are performance-oriented; performance being the primary impetus in using RISC technology. Because speed equals power and RISC machines are high-speed, how can we reduce power without sacrificing performance?

Let's examine the memory system. A RISC memory system consists of two components — the main memory and the cache. The main memory, DRAM or SRAM, is only accessed when a cache miss occurs. If main memory is SRAM, it can be deselected effectively by putting the SRAM in power down mode. If the main memory is DRAM, it consumes little power outside of refresh and access. We can reduce the instantaneous current demands by using different refresh methods.

During refresh, a large current spike is needed to revitalize the charge in the storage capacitors. The fewer DRAMs you refresh at a time, the lower instantaneous current the system will consume. To lower power consumption, the designer can distribute the refresh cycles over time and over banks of DRAM memory. It is important to note that, although this increases the complexity, it rarely impacts performance. Many DRAM controllers on the market today use this method of reducing DRAM power. This design philosophy should also reduce the noise a system produces.

The cache memory subsystem is constantly accessed and, therefore, power reduction here is more complex. To better understand the power reduction opportunities, a more detailed understanding of this subsystem is needed.

As in all RISC engines the R3000 implements a local cache. The Harvard architecture cache structure of the R3000 is divided into two components; the instruction cache and the data cache. Both caches are physically mapped so that they can include their respective data, tag, valid bit, and parity equaling 60 bits per line (a line is a cache entry that has a unique tag).

During a single R3000 cycle, both the instruction and data cache are accessed. If the RISC engine is running at 33MHz, the caches run at 264 MBytes/sec. This cache scheme requires that the cache be continuously selected, therefore always powered up. The only way power can be saved in the cache subsystem is by deselecting the caches during a cache miss.

A model will help us understand how much power can be saved. Assume an average program contains the following characteristics: 55% ALU instructions, 15% branch instructions, 20% load instructions, 10% store instructions, 95% I-cache hit rate, 85% D-cache hit rate, cache missing read/write latency = 5 cycles. If we were to use a program that has 100 instructions we could compute:

55 ALU instr. @ 1 cyc./instr. =	55.0 cyc.
10 store instr. @ 1 cyc./instr. =	10.0 cyc.
15 branch instr. @ 1.5* cyc./instr. =	22.5 cyc.
20 load instr. @ 1.3* cyc./instr. =	26.0 cyc.
Total =	113.5 cyc.

*Since the R3000 has branch and load delay slots, we are going to *conservatively* estimate that the compiler can fill branch delay slots 50 percent of the time with useful instructions and fill load delay slots 70 percent of the time with useful instructions. Unfilled delay slots have nops inserted which do not count as useful instructions.

This means that a 100 instruction program takes 113.5 instructions when we take into account the load and branch delay slots. This results in a cycle instruction rate of 113.5/100 = 1.35 cycles per instruction. The cache miss rate is then added to the above rate.

Instruction Cache Performance @ 95% Hit Ratio

95 instructions hit in cache *1.135 cycles/instruction + 5 Missed instructions *5 cycles latency *1.135 cycles/instruction = 136.2 cycles.

Data Cache Performance @ 85% Hit Ratio

20 total data loads which miss at 15% rate
20 * .15 = 3 load instructions miss
3 loads @ 5 latency cycles = 15 cycles

As shown, our original 100 instruction program, which took 113.5 cycles without cache hit criteria, now takes a total of 136.2 + 15 = 151.2 cycles. Since 5 instructions missed and 3 loads missed, 40 out of 151 were cache idle cycles or approximately 25%. Deselecting the cache during this time will produce the power consumed by the cache memory system.

When implemented this scheme, keep in mind that when CS = False the SRAM goes into a standby node. One could assume that the power savings could be 87% for 25% of the time. With any method used to generate the percentages, the savings will be measurable.

It is very easy to decode real stalls and deselect the cache — reference Figures 1 and 2 Single Word Transfer (included). At the beginning of a stall, the R3000 output MemRd goes low to signal a cache miss and request a main memory read cycle. Since the RdBusy input is normally high, the boolean equation sets Cache CS high, deselecting the caches. Since RdBusy is sampled at the end of phase one, a complete cycle exists before the caches are needed. The R3000 then sets $\overline{\text{MemRd}}$ high at the beginning of the fix-up cycle. After the fix-up cycle is complete, the main memory controller returns the RdBusy high again in preparation for the next miss cycle. A simple boolean equation can be written to provide the deselection function:

Cache-CS = $\overline{\text{MemRd}}$ * RdBusy

Write stalls are just as easy to decode. Note that write stalls only occur when the write buffer is full — s.e., data cannot be written into main memory at the processor rate. Therefore, because the model does not include write buffer stall cycles, their help will be difficult to demonstrate. Simulation history shows that these cycles are a small amount of the the total stall cycles, but we show how to decode them and deselect the cache since it takes very little logic to do so.

$\overline{\text{MemWr}}$ goes low to indicate when a main memory write must be performed. Conventional write buffer operation will set $\overline{\text{WrBusy}}$ low after it is full in order to stall the R3000 on the next main memory write cycle. The R3000 will stall if another Store cycle occurs while the write buffer is full and WrBusy is still low. Note that during the stall, $\overline{\text{MemWr}}$ goes false so it must be captured with a register clock on $\overline{\text{SysOut}}$. When the data can be accepted, $\overline{\text{WrBusy}}$ can be set high to terminate the stall. One cycle later, a fix-up cycle will occur where the data can be captured in phase two. With this in mind, Cache-CS is NOT $\overline{\text{WrBusy}}$.

SUMMARY

The R3000 has features that can minimize the power consumed. The model program created does not take into account instruction streaming or block refill. These features will improve performance and cache utilization which is inversely proportional to idle cache time. Because performance is paramount and these features require added hardware, tradeoffs become more difficult. In the final analysis, cache idle time is a percentage of operating time — that percentage being application-specific. It is up to the designer to trade off the additional gates with power savings which can be significant.

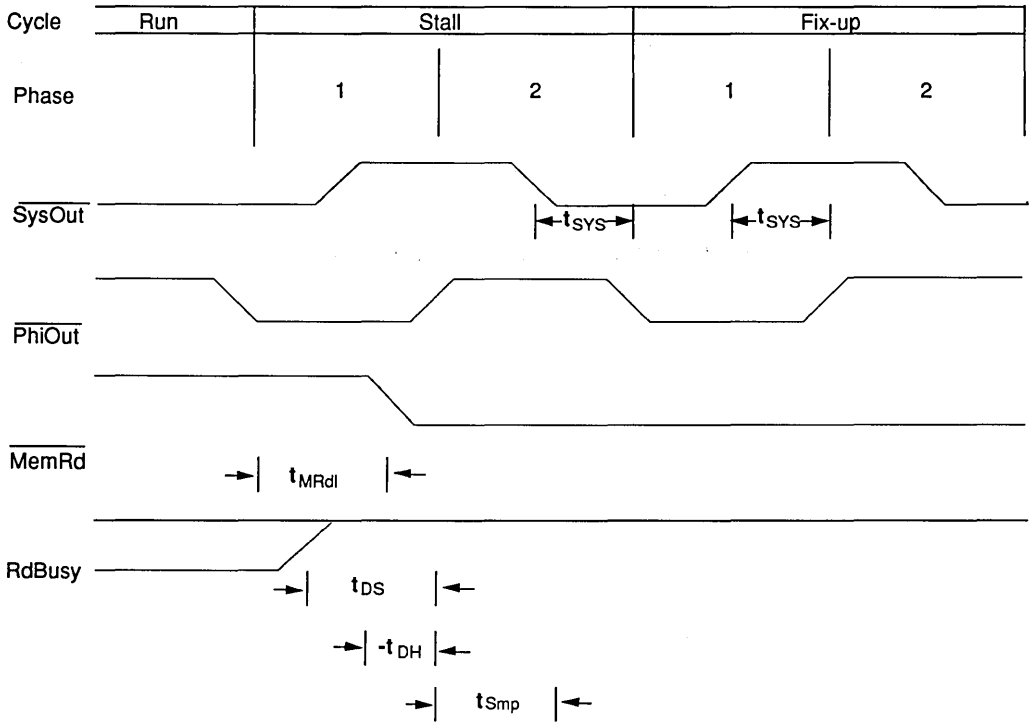


Figure 1. Single Word Instruction Transfer — Beginning of Stall

9

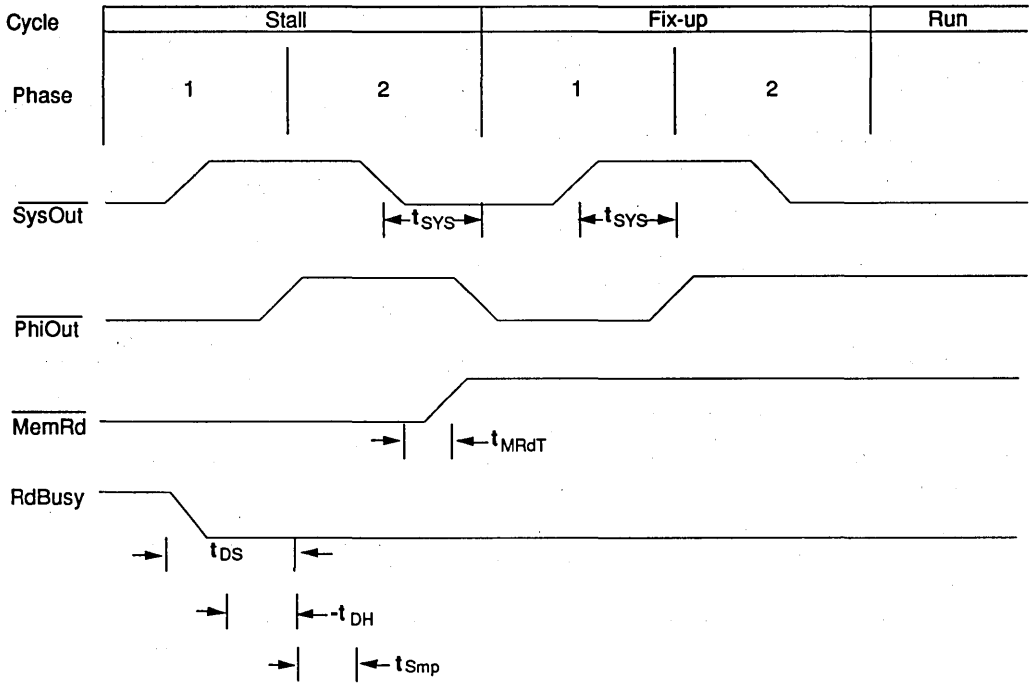


Figure 2. Single Word Instruction Transfer — End of Stall

Integrated Device Technology, Inc.

PARITY TRADE-OFFS WITH THE R3001 RISController™

APPLICATION NOTE AN–58

by Philip Bourekas

The R3000 includes parity over the data and tag portions of both the instruction and data caches. While this gives the system reliability people a "good" feeling about the reliability of the design, a detailed analysis of the cost versus the benefits of parity protection may lead the system architects to a different decision.

The IDT R3001 RISController™ removes the restriction that parity must be used, and allows the system designer full flexibility to make the cost/benefits trade-offs. This application note provides information to help this analysis.

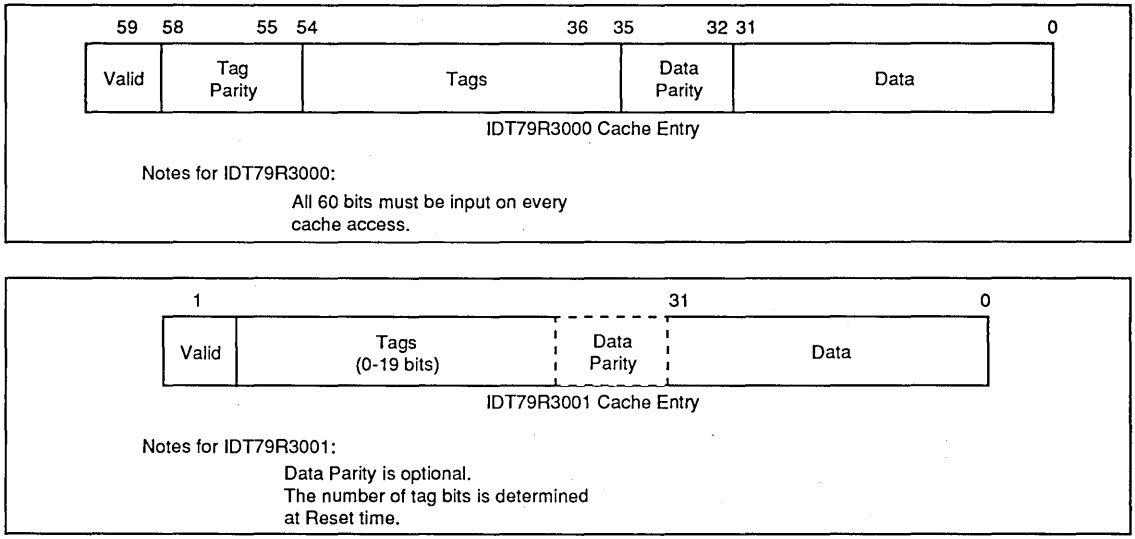


Figure 1. Cache Line Formats of R3000 and R3001

2527 drw 01

THE COST OF PARITY

Parity in the system caches is costly and occurs in a number of areas:

Static RAM Cost

Parity consumes 7 bits from each of the instruction and data caches, for a total of 14 bits of cache. In most workstation applications parity adds to the system cost a minimum of 4 high-speed 16K x 4 static RAM devices.

Figure 1 illustrates the different cache formats of the R3000 and the R3001.

Read/Write Buffer Interface Logic

Parity causes the system bus to be widened by 4 bits, thereby requiring extra devices in the read/write buffer path to accommodate parity. Additionally, typical designs do not carry parity in main memory: computer systems use EDC (Error Detection and Correction); embedded systems do not choose to incurr the overhead, cost and complexity of parity in main memory and, thus, do not include either parity or EDC. The requirement of parity in the caches also means that these systems must provide for parity generation logic using, as an example, four IDT74F280 parity generators.

The shaded portions of Figure 2 illustrate the added overhead of parity in R3000 systems.

Memory Speed

In systems which omit parity from main memory, there is a performance impact for requiring cache parity. Block refill of the R3000 relies on data coming into the cache at the rate of one word per CPU clock cycle. With a parity generator in this path, the system designer must either use faster memory to allow time for the parity generators (memory must be fast enough to both provide a new word and calculate its parity at the rate of the CPU clock cycle), or the designer must pipeline memory. This requires adding extra logic and an extra cycle of latency to main memory so that the parity generation logic has a full clock cycle to produce results. Thus, the system designer is faced with two unattractive alternatives: purchase faster (more expensive) main memory devices than would otherwise be required, or degrade performance and purchase extra logic to pipeline main memory to give the parity generation logic ample time. This is illustrated as the pipeline stage in Figure 2.

9

Cache Design Complexity

Many embedded applications utilize techniques (discussed in other IDT application notes) which reduce the number of tag bits that must be stored in the cache memories. However, the R3000 still requires that parity be provided to the processor on every cache access; this significantly complicates the design of these reduced systems since parity must be stored over tag bits which may not actually reside in memory.

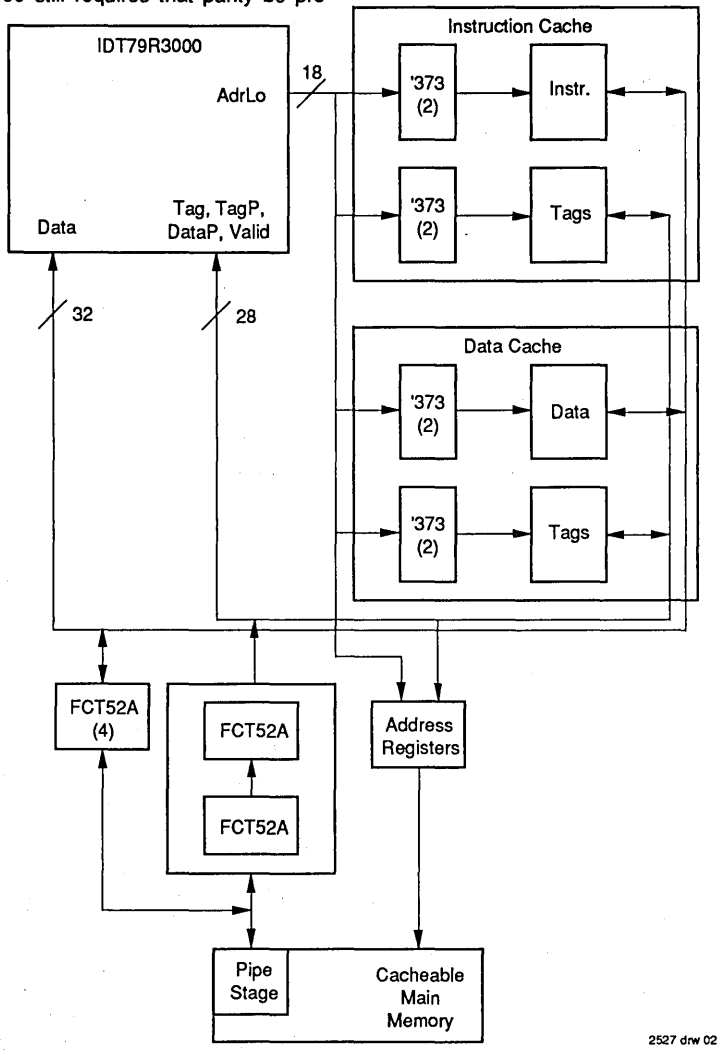


Figure 2. Impact of Parity on Main Memory and Interface

THE BENEFITS OF CACHE PARITY

Now that the full cost of cache parity is understood, it is important to recognize the protection gained by parity.

Cache parity is not intended to uncover timing or electrical faults in the design. It is also typically not intended to uncover hard failure devices. Instead, parity is designed to detect the occurrence of a "soft failure", whereby a "good" RAM has the value in one of its cells flipped by some external phenomenon, typically alpha particles. When one of these events occurs the parity over the value which was changed would indicate that something was wrong and it is left to the system to decide how to deal with the event. In the case of the R3000, it is dealt with by processing a cache miss. In the R3001, when data parity is enabled, a cache miss is processed as in the standard R3000 and, additionally, the parity error is reported on the $\overline{\text{PErr}}$ pin.

In the R3000 there are two separate areas of each cache which are protected by parity. These are the tag and the data areas. Parity actually benefits the data portion of the system much more than it does the tag. If a tag bit flips due to a soft error, then to have an adverse effect, it must flip from a value that would have caused a cache miss to a value that allows a cache hit. Otherwise, a cache miss would be processed in any case. The incremental system reliability of TAG parity is very minimal since the statistical likelihood of a soft error forcing the incorrect instruction to be processed is minimal.

Parity over the data portion of the caches performs a more important function. If a soft error were to occur, the consequences would be more serious: an erroneous data value may be loaded, an erroneous instruction may be executed or an illegal instruction exception may occur. Parity over this 32-bit field does have a finite, measurable benefit.

The system designer must understand just how big is that benefit and weigh it against the system cost of maintaining parity. If the likelihood of a cache parity error is statistically insignificant relative to the probability of a disk failure or some other device failure, then the cost of parity far outweighs any benefit.

The industry standard measure of the probability of a soft error is the FIT rate of the RAM devices. One FIT is equivalent to a mean time-to-failure of 10^9 device hours of operation. The FIT rate is measured by accelerating the "typical" life of the product; typically, soft errors are accelerated by exposing the die to Thorium–232.

IDT static RAMs exhibit FIT rates of less than 10 (polyimide die coating, used on most IDT SRAMs, lowers the FIT rate to effectively 0). This results in a mean time-to-failure of a single RAM device longer than 100 million device hours. Of course, the more RAMs you use, the "less reliable" the system and the shorter the mean time-to-failure. Typically, as the number of devices increases, an RMS analysis is used to determine the decrease in reliability This analysis will use the more severe assumption that n devices cause the reliability to degrade to 1/n of what it is with one device.

If each 32-bit cache is built from eight 4-bit wide static RAMs, then there are 16 devices (this is the worst case; wider RAM devices have equivalent FIT rates, but fewer devices are required to build the caches). For mathematical simplicity, let us use a FIT rate of 6.67 for each RAM (this FIT rate is significantly higher than that measured for the IDT 16K x 4 high-speed SRAM). Thus, 16 devices would have (16 x 6.67) failures in 10^9 hours, or a mean time-to-failure of 10^7 hours.

Ten million hours is over 400,000 days or over 1,100 years. Thus, providing parity for the R3000 cache is designed to safeguard for an event which is not likely to occur until after the year 3090.

WHY DOES ANYONE USE PARITY?

Given that parity appears extremely costly relative to the likelihood of the problem it solves, why does anyone go to the trouble of using it? This question has a number of answers.

First, not all memory device architectures have such high resistance to soft errors. DRAM devices typically involve smaller, single transistor cells with no active pull-up device to lock a value into the data cell, while SRAM devices require 4 to 6 transistors per memory bit and actively pull-up logic high values in the cell (Figure 3). DRAM devices require that each memory cell be periodically refreshed in order to maintain its stored value, while SRAMs have no such requirement. Thus, DRAMs are considerably more prone to soft errors than SRAMs, and parity protection is sometimes employed in these systems. Studies indicate that main memory, composed of extensive banks of DRAM, can have a probability of soft error in as short a period as 20 days. Therefore, more designs are moving to Error Detection and Correction to protect main memory.

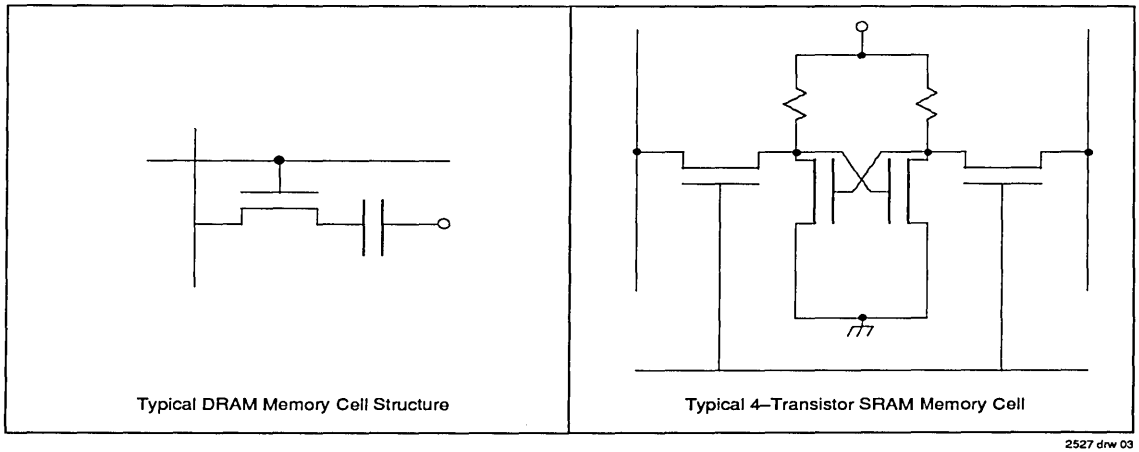


Figure 3. DRAM Memory Cell Structure and SRAM Memory Cell Structure

Even within systems that use static RAM devices, parity may still be an acceptable cost. Space applications would tend to expose memory to a higher incidence of alpha–particles and, thus, have a higher likelihood of soft errors. Certain commercial applications prefer to support parity despite its cost. Banking applications, such as on-line transaction processing systems for ATM networks, could involve a significantly higher cost if a data error was allowed to propagate through the system. Other systems promote system parity as a "marketing" feature, promoting its "benefits" as added value to the customer.

PARITY ON THE R3001

The R3001 treats parity differently than the standard R3000. First of all, TAG parity is not supported at all. As shown above, the probability of a system problem due to a soft error on a RAM containing TAGs is extremely small. Since the R3001 supports variable TAG bus width, implementing TAG parity over this variable width field would have significantly complicated the control logic and slowed overall device operation.

On the data portions of each cache, the system designer has the ability to decide whether or not a given system supports data parity. If the application desires parity, the system designer can make this design decision. In this application, the system would still benefit from features of the R3001 such as its ability to reduce the width of the TAG bus (and thus reduce system cost) Another benefit from the R3001 is signaling to the system that a parity error has occurred and therefore allowing the system to distinguish this event from a normal cache miss cycle.

SUMMARY

The R3001 RISController™ gives the system designer total flexibility over the cost/benefits trade-offs of the application. For the majority of systems and embedded applications, parity can be omitted and significant cost savings can be achieved, For systems which still require parity, the R3001 allows the system designer to maintain parity while still adding features and benefits not found in the standard R3000. The result is a device well suited to a variety of applications, and software compatible (both at the kernel and user level) with the popular R3000.

Integrated Device Technology, Inc.

IDT79R3000 33MHz SPECIFICATION AND CACHE TIMING

APPLICATION NOTE AN-61

by Satyanarayana Simha, Michael Miller, David Winn, Philip Bourekas

INTRODUCTION

The reduced instruction set computer, IDT79R3000, has allowed for simplicity in hardware and a synergistic relationship between the architecture and compilers. To increase the throughput of an IDT79R3000-based system, direct mapped cache memory is implemented. The availability of high-speed static RAMs from IDT gives the designer the flexibility of selecting the proper part for his cache system. To select the proper RAM to function as a cache for the IDT79R3000, it is necessary to know the speed requirements for the RAM. This application note contains timing information pertaining to the IDT79R3000 operating at different frequencies and those of the static RAMs that function as cache. The timing for the cache RAMs (ordinary SRAMs) was derived from the IDT79R3000 requirements and by making certain assumptions. This document clearly details those assumptions and the methodology used to calculate the RAM AC timing parameters. This methodology can be used by the designer to deduce the RAM timing parameter requirements for the IDT79R3000 operating at any frequency.

GENERAL DESCRIPTION

The IDT79R3000 is a RISC microprocessor used in a variety of applications ranging from low-end embedded controllers to high-end workstations. Until recently, the IDT79R3000 operated at a frequency of up to 25MHz. The need for more performance necessitated that the IDT79R3000 operate at higher frequencies. Therefore, this application note specifies the timing considerations for an IDT79R3000 microprocessor operating at 33MHz. This document does not explain the functionality of the IDT79R3000 nor its architecture, but is limited to describing the key timing parameters. For a more detailed description of the functionality and an architectural description of the IDT79R3000, please refer to the references listed at the end of this document.

This application note starts with a brief description of the IDT79R3000, the four-phase clock inputs, cache timings, required timings for SRAMs to function as cache for the IDT79R3000, the factors used in the timing calculations, and the conclusion reached.

FUNCTIONAL DESCRIPTION

The IDT79R3000 is a 32-bit RISC microprocessor that is currently available from Integrated Device Technology, Inc. in two packages: the 144-pin PGA and the 172-pin ceramic flatpack. It has a 32-bit data bus, a 32-bit address bus that is divided into the low-order 16-bits (AdrLo) and the high-order 16 bits (AdrHi), control signals for the cache, control signals for main memory, and power and ground pins. The IDT79R3000 also has four double-frequency clock inputs and one clock output used for interfacing the IDT79R3000 to the external world.

FOUR-PHASE CLOCKS AND DELAY-LINE SETTINGS

Figure 1 shows a block level diagram of the IDT79R3000 with its four clock inputs coming from a delay line. Table 1 shows a summary of the delay line settings to be used for different operating frequencies of the IDT79R3000. **Please note carefully that Clk2xSys is taken as the zero time reference and comes from the first tap of the delay line.** The other 2x clocks lag Clk2xSys in time and follow it with respect to delay line taps.

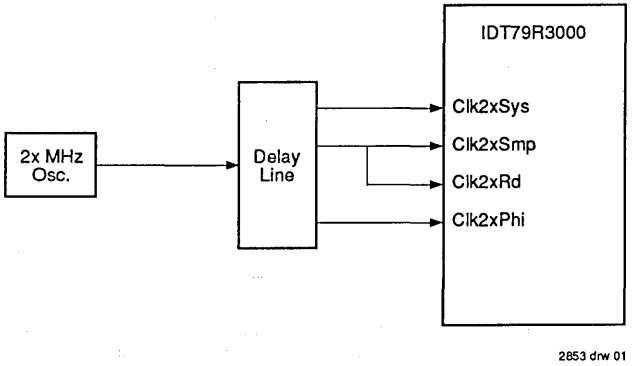


Figure 1. Four-Phase Clock Input to the IDT79R3000

9

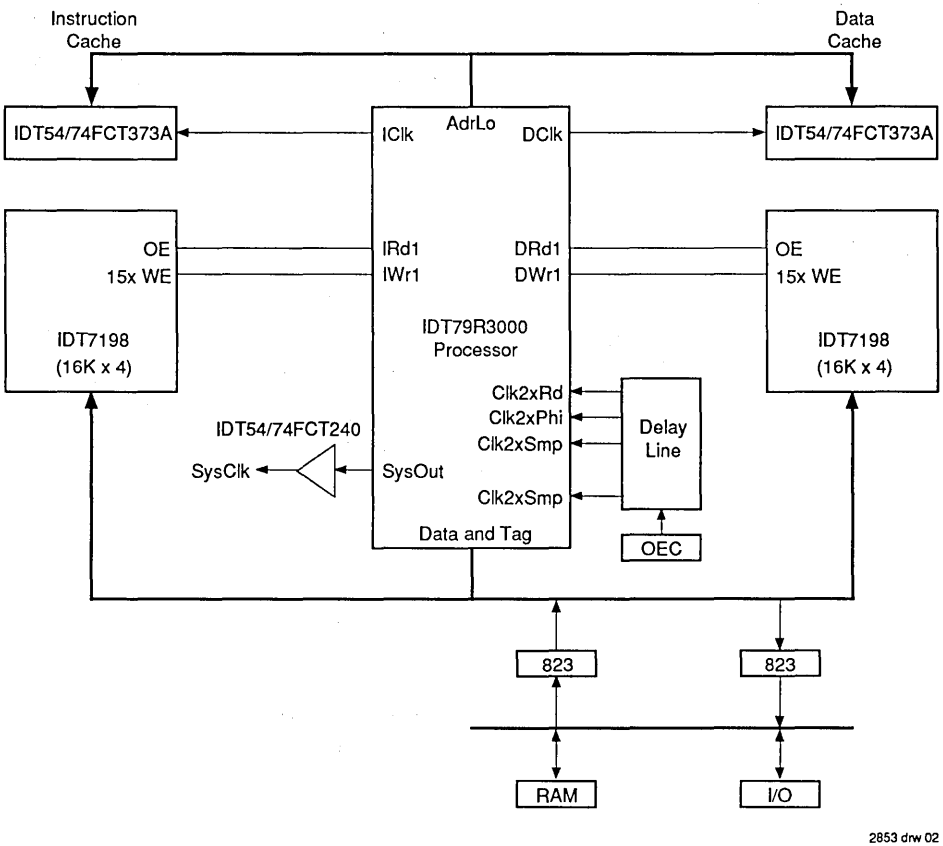
Parameter	16MHz	20MHz	25MHz	33MHz
Clk2xSys	0	0	0	0
Clk2xRd	6	6	6	4.5
Clk2xSmp	6	6	6	4.5
Clk2xPhi	16	14	12	9

2853 tbl 01

Table 1. Delay Line Settings for IDT79R3000 Operating at Different Frequencies

DERATING CALCULATIONS AND CACHE TIMING CONSIDERATIONS USING X 4 SRAMS

The design of the cache subsystem for the IDT79R3000 is straightforward. Industry standard static RAMs function as cache. This section discusses the methodology used to calculate the critical timing parameters for a static RAM so that it can function as cache for the IDT79R3000. This section also examines the timings for a 16MHz, 20MHz, 25MHz and 33MHz IDT79R3000. The timing equations derived take into account the effect of capacitive loading on the bus. The derating factors are calculated based on certain assumptions. These assumptions are detailed in this section and the derating factors are calculated. The timing equations are then discussed. At the end of this section, a table is included containing the SRAM timings for different operating frequencies of the IDT79R3000.



2853 drw 02

Figure 2. Block Level Diagram of a Cache Subsystem With the IDT79R3000 Using IDT7198 16K x 4 to Function as Cache (the IDT79R3010 is not shown in this figure)

Device Capacitance

Figure 2 shows a typical IDT79R3000-based system. The cache is comprised of fast 16 K x 4 static RAMs (i.e., the IDT7198). The AdrLo bus of the IDT79R3000 goes through a latch: the IDT54/74FCT373. It also goes through a latch which is used to address the main memory. All the devices have an input and an output capacitance. In addition, each device is capable of driving a certain load. These parameters: the input capacitance, the output capacitance and the load capacitance, are given in Table 2.

The cache format of the IDT79R3000 is comprised of 60 bits — 32 bits of data, 4 bits of data parity, 20 bits of tag, a valid bit, and three bits of parity to cover the tag and the valid bit. With this requirement, it is clear that, for the instruction cache, fifteen IDT7198s (16K x 4 SRAMs) are needed. The same number is also required for the data cache. This means that there are a total of 30 SRAM devices for the cache.

Device	Number of Devices	Capacitance	Total Capacitance
IDT79R300	1	C_{IN} = 10pF	10pF
IDT7198	2	C_{OUT} = 7pF	14pF
IDT374A	1	C_{OUT} = 12pF	12pF
IDT823B	1	C_{IN} = 10pF	10pF

2853 tbl 02

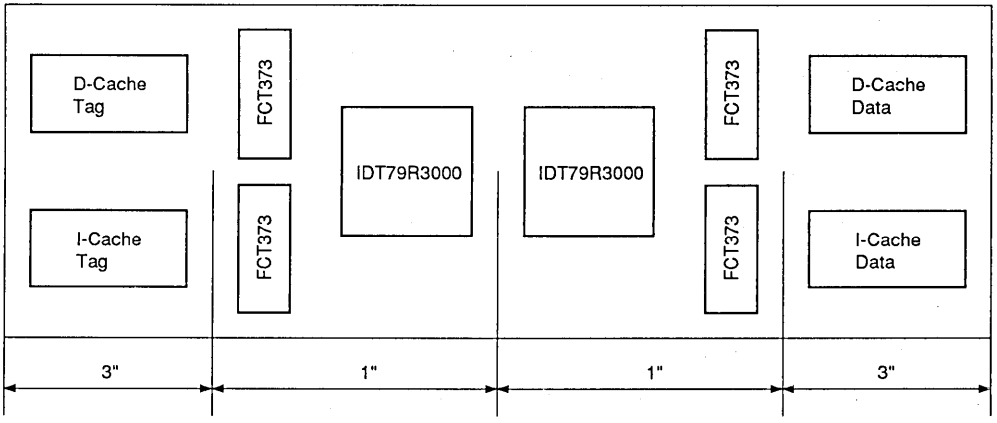
Table 2. Capacitances of the Various Devices in a Typical IDT79R300 Systems

Assumptions for Surface Mount Layout Design with x4 SRAMs

In the following sections, certain assumptions have been made while calculating the derating factors. These are as follows:

1) The trace has a capacitance of 2pF/inch.
2) The speed of light is 2ns/foot in epoxy.
3) The IDT79R3000 speeds are specified with a loading of 25pF. For every additional 25pF, there is a delay of 1ns.
4) The distances between the IDT79R3000 and the latches are approximately 1inch each.
5) The distances between the IDT79R3000 and the RAMs are approximately 4inches each.
6) In all of the assumption, it is assumed that a surface mount package is used. Figure 3 shows a brief mechanical layout of an IDT79R3000 board.

Figure 3 shows a brief mechanical layout of an IDT79R3000 board.



Assume read and write buffer underneath the board

2853 drw 03

Figure 3. Surface Mount Board of an IDT79R3000 System With Cache and Main Memory Interface and Approximate Distances Between the Various Devices

9

Address Bus Derating Calculations

For the system shown in Figure 4.1, each address bit is connected to five latches: one going to the main memory interface buffer, two to the instruction cache and tag memory, and two to the data cache and tag memory, respectively. The latches in turn are connected to the address pins on the static RAM. Figure 4 shows all the devices to which each address bit is connected.

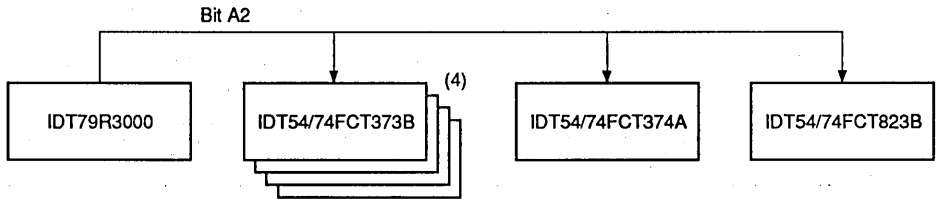


Figure 4. Block Diagram Showing Various Devices Connected to One Address Bit

Trace length from the CPU to the latch = 4inches — (1)

Capacitance of the trace, C_{trace}= 4 x 2pF/inch = 8pF — (2)

Input capacitance of the 373 latch = 10pF — (3)

As each address bit is connected to five latches (IDT54/74FCT373),
Total input capacitance due to 5 devices, C_{373in} = 5 x 10 = 50pF — (4)

Total capacitive load = C_{trace} + C_{373in} = 8 + 50 = 58pF — (5)

The rated IDT79R3000 load, $C_{L(R3000)}$ = 25pF — (6)

From Eq. (5) and Eq. (6),
Extra capacitive loading for the IDT79R3000 = 58 - 25 = 33pF — (7)

Let us now examine the capacitive loading between the latches and the RAM.
Path length from latches (IDT54/74FCT373s) to RAM (IDT7198s) = 3" — (8)

Trace capacitance from latch to RAM = 3 x 2pF/in = 6pF _ (9)

Input capacitance of the RAM = 5pF — (10)

Each output from the latch is connected to eight RAM devices.
Load due to 8 devices = 8 x 5 = 40pF —(11)

Total capacitance = 40 + 6 = 46pF — (12)

The rated 373 load = 50pF — (13)

From Eq. (12) and Eq. (13) it can be seen that there is no delay due to the capacitive load between the latch and the RAM. However, there is a delay due to the capacitive load between the IDT79R3000 and the latch. This delay can be calculated as follows:

For every extra 25pF of load, there is a delay of 1ns. — (14)

From Eq. (7) and Eq. (14), delay due to the capacitive load = 33 / 25 = 1.32ns - (15)

The speed of light ≈ 2ns/foot — (16)

For a maximum path length of 5", delay = 5"/12" x 2 = 0.8ns — (17)

From Eq. (15) and Eq. (17),
Total propagation delay for the address bus, $AdrLo^d$ = 1.32 + 0.8 ≈ 2ns — (18)

Data Bus Derating Calculations

The derating calculations for the data path are similar to those done for the address path. The data bus is connected to the floating point unit (IDT79R3010), the instruction cache (IDT7198), the data cache (IDT7198), a read register (IDT54/74FCT374A) and a write register (IDT54/74FCT823). This is shown in Figure 5. Two cases must be considered: a data store and a data fetch. Both are discussed.

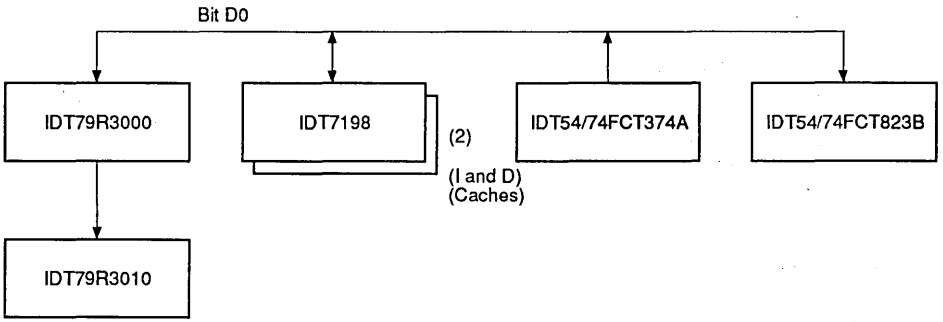


Figure 5. Block Diagram Showing Various Devices Connected to One Address Bit

Data Store (IDT79R3000 CPU Outputs Data):

Each data bit is connected to two RAMs (IDT7198s) —one for instruction and one for data.

The path length for the data bus = 5" — (18)

Trace capacitance for the data bus = 5 x 2pF/in = 10pF — (19)

Capacitive loading due to devices,
$C_{devices} = 2 \times C_{RAMin} + C_{374in} + C_{823} + C_{R3010}$ — (20)

$C_{devices} = 2 \times 7 + 12 + 10 + 10 = 46pF$ — (21)

Total capacitive load = $C_{trace} + C_{devices} = 46 + 10 = 56pF$ — (22)

Propagation delay due to speed of light = 5"/12" x 2 = 0.8ns — (23)

Delay due to capacitive load = (56 -25) /25 = 1.24ns — (24)

From Eq. (23) and Eq. (24),
Total propagation delay on a store = 1.24 + 0.8 ≈ 2ns — (25)

Load (RAM Provides Data)

Since the trace length is the same, C_{trace} = 10pF _ (26)

Capacitive load due to devices,
$C_{devices} = C_{R3000} + C_{R3010} + C_{RAMin} + C_{374in} + C_{823}$

$C_{devices} = 10 + 10 + 12 + 10 + 7 = 49pF$ — (27)

9

Total capacitance = $C_{trace} + C_{devices}$ = 10 + 49 = 59pF

The RAM rated drive is 30pF

Extra load = Total capacitance - RAM rated drive = 59 - 30 = 29pF — (28)

Propagation delay due to capacitive load = 29/25 = 1.16ns — (29)

Propagation delay due to the path length = 0.8ns — (30)

Total propagation delay = 1.16 + 0.8 ≈ 2ns — (31)

Read and Write Control Derating Calculations

The effect of the capacitance on the control signals from the IDT79R3000 processor to the caches and the memory interface is considered here. The control signals on the IDT79R3000 are the $\overline{IRd}$, $\overline{DRd}$, $\overline{IWr}$ and $\overline{DWr}$ which control the instruction cache read, data cache read, instruction cache write and data cache write, respectively. The read and write control signals are connected to the output enable ($\overline{OE}$), and write enable ($\overline{WE}$) of the instruction and data cache, respectively. Two control signals each are provided for the read and write operations of each of the caches. Assuming the use of a 16 K x 4 IDT7198 static RAM, each control signal is connected to eight such static RAMs.

Number of devices (SRAM) connected to each control line = 8 — (32)

Input capacitance of each device (SRAM) = 5pF — (33)

Total load capacitance = 5 x 8 = 40pF — (34)

Path length = 5" — (35)

Trace Capacitance = 5 x 2pF/in = 10pF = 10pF — (36)

Total capacitance = 40 + 10 = 50pF — (37)

Extra capacitive load = 50 - 25 = 25pF — (38)

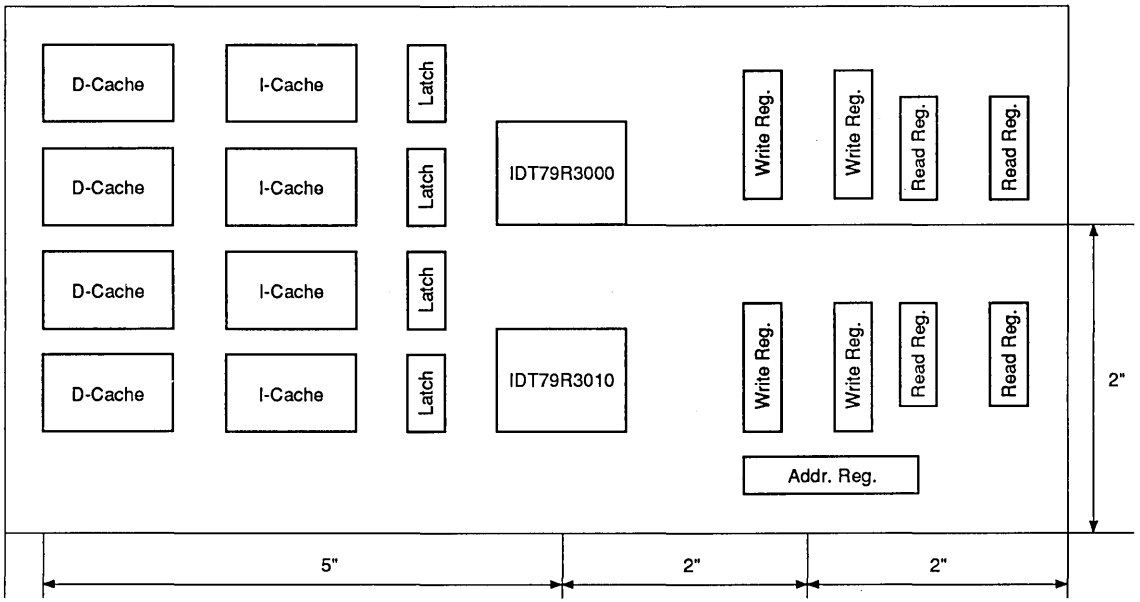
Propagation delay due to capacitive load = 1ns — (39)

Propagation delay due to the trace length = 0.8ns — (40)

Total propagation delay = 1 + 0.8 ≈ 2ns — (41)

Assumptions for Through-hole Layout Design Using x4 SRAMS

In this section, the deratings are calculated for a through-hole design. Figure 6 shows an example of the layout of a through-hole design. This layout corresponds to an IDT79R3000 demonstration board used extensively at IDT. The data trace lengths are 10 inches and the address trace lengths are 9 inches.



2853 drw 06

Figure 6. Board Layout for a Through Hole Design of an IDT79R3000 Cache Subsystem

Address Derating Calculations

For the system shown in Figure 4, the number of devices connected to the IDT79R3000 is the same.

Trace length from the CPU to the latch = 9" — (42)

Trace capacitance = C_{trace} = 9 x 2 = 18pF — (43)

Input capacitance of the latches = 10pF — (44)

Total capacitance = 5 x C_{373} + C_{trace} = 5 x 10 + 18 = 68pF — (45)

Extra load on the R3000 = C_L = 68 - 25 = 43pF — (46)

Since the rated IDT54/74FCT373 load is 50pF, there is no derating factor between the IDT54/74FCT373 and the RAMs.

Therefore the derating is between the IDT79R3000 and the latches.

Delay due to capacitance = 43/25 = 1.75ns — (47)

Propagation delay due to the trace length = 9/12 x 2 = 1.5ns — (48)

Total derating on the address bus = 1.75 + 1.5 ≈ 3ns — (49)

Derating on the Data Bus

As in the section entitled **Data Bus Derating Calculations**, the derating for the data bus is calculated for two cases: i) an instruction fetch and ii) data store.

Data Store (IDT79R3000 CPU Outputs Data)

Each data bit is connected to two RAMs (IDT7198s) — one for instruction and one for data.

9

The path length for the data bus = 10" — (50)

Trace capacitance for the data bus = 10 x 2pF/in = 20pF — (51)

Capacitive loading due to devices,
$C_{devices} = 2 \times C_{RAMin} + C_{374in} + C_{823} + C_{R3010}$ — (52)

$C_{devices} = 2 \times 7 + 12 + 10 + 10 = 46pF$ — (53)

Total capacitive load = $C_{trace} + C_{devices} = 20 + 46 = 66pF$ — (54)

Propagation delay due to speed of light = 10"/12" x 2 = 1.6ns — (55)

Delay due to capacitive load = (66 - 25) /25 = 1.55ns —(56)

From Eq. (23) and Eq. (24),
Total propagation delay on a store = 1.54 + 1.67 ≈ 3ns — (57)

Load (RAM Provides Data)
Since the trace length is the same, C_{trace} = 20pF — (58)

Capacitive load due to devices,
$C_{devices} = C_{R3000} + C_{R3010} + C_{RAMin} + C_{374in} + C_{823}$

$C_{devices} = 10 + 10 + 12 + 10 + 7 = 49pF$ — (59)

Total capacitance = $C_{trace} + C_{devices} = 20 + 49 = 69pF$

The RAM rated drive is 30pF

Extra load = Total capacitance - RAM rated drive = 69 - 30 = 39pF — (60)

Propagation delay due to capacitive load = 39/30 = 1.3ns — (61)

Propagation delay due to the path length = 1.6ns — (62)

Total propagation delay = 1.3 + 1.6 ≈ 3ns — (63)

Read and Write Control Deratings

For a through-hole design, the effect of derating on the control signals will be more. This section calculates that effect. The trace length from the CPU to the RAMs is 9 inches for the layout shown in Figure 6. Each control signal is connected to 8 devices.

Number of RAM devices connected to each control signal = 8 — (64)

Input capacitance of each RAM = 5pF — (65)

Total load capacitance = 8 x 5 = 40pF — (66)

The trace length = 9' — (67)

Trace capacitance = 9 x 2pF/inch = 18pF — (68)

Total load capacitance = 40 + 18 = 58pF — (69)

Extra load = 58 - 25 = 33pF — (70)

Derating due to capacitive load = 33 / 25 = 1.35ns — (71)

Propagation delay due to trace length = 9 / 2 x 2ns/foot = 1.5ns — (72)

Total derating = 1.35 + 1.5 ≈ 3ns — (73)

Timing Equations for Cache Design

This section deals with the timing equations that enable us to determine the critical timing requirements of the static RAM that will be used as cache. These equations are based on the use of static RAMs (without built-in latches) as cache RAMs. The superscript 'd' in the following equations denotes the deratings to be taken into account. The static RAM chosen for illustration here is a 16K x 4 IDT7198. **The board is assumed to be surface mount for all speeds of the IDT79R3000 except for the 16MHz speed grade.** The deratings for the surface mount board are 2ns and 3ns for a through-hole board (which is used for the 16MHz IDT79R3000). The deratings were derived from certain assumptions. The explanation and the methodology used is set forth in the previous sections. Following is a generalized equation and the timing requirements for different frequencies of the IDT79R3000. All calculations are based on the IDT79R3000 specifications for the four speed versions (16, 20, 25 and 33MHz), which are found in the IDT data sheets.

Figures 7, 8, 9, and 10 show the timing diagrams of the IDT79R3000 when it is doing a data store followed by an instruction fetch. This is the worst case example and is chosen to determine the SRAM parameter requirements. Figure 7 shows the timing diagrams for an IDT79R3000 operating at 16MHz. Figures 8, 9 and 10 show the timing diagrams for an IDT79R3000 operating at 20MHz, 25MHz and 33MHz, respectively. The encircled numbers represent the equations presented in the section entitled **Timing Equations for Cache Design.** The timing diagram, in conjunction with the equations, are used to arrive at determining the timing requirements.

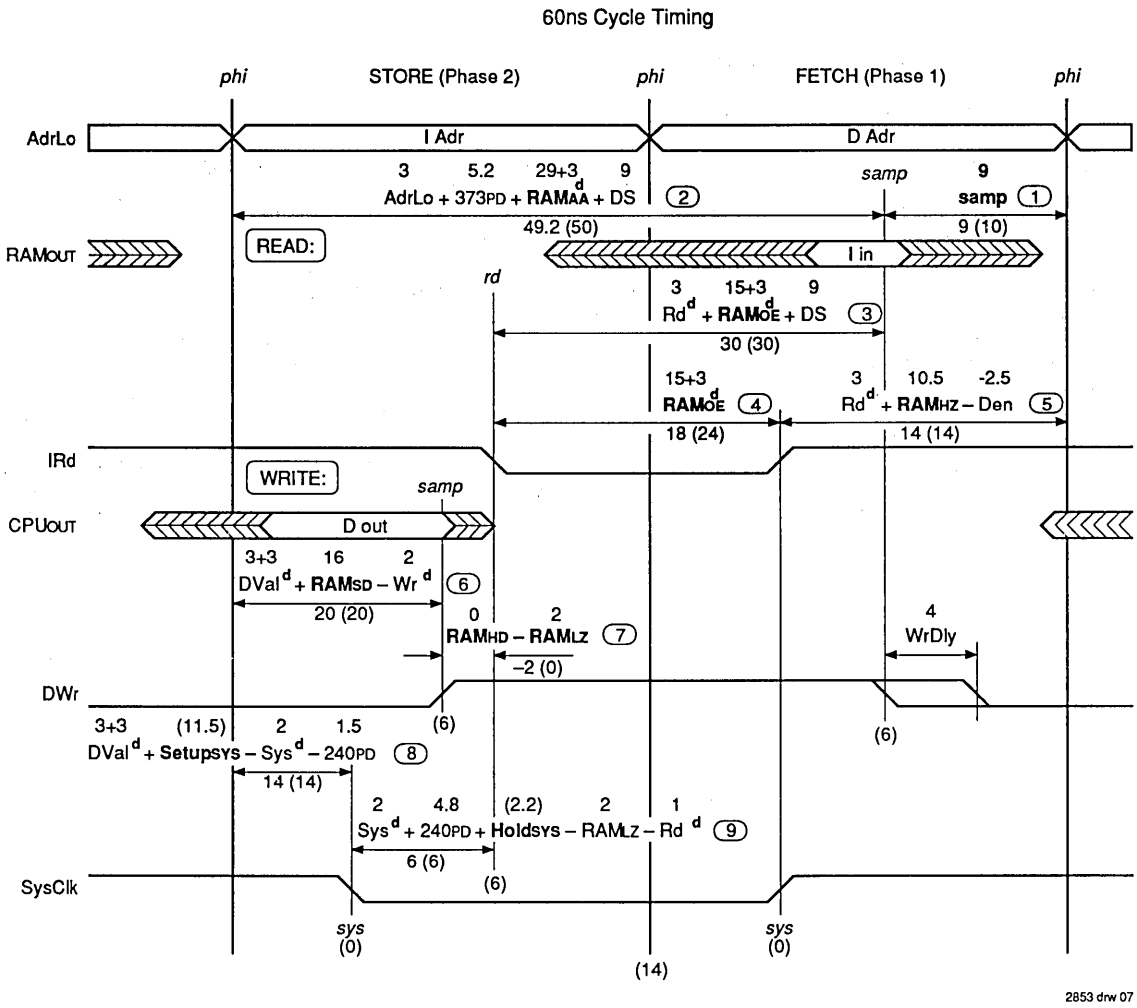
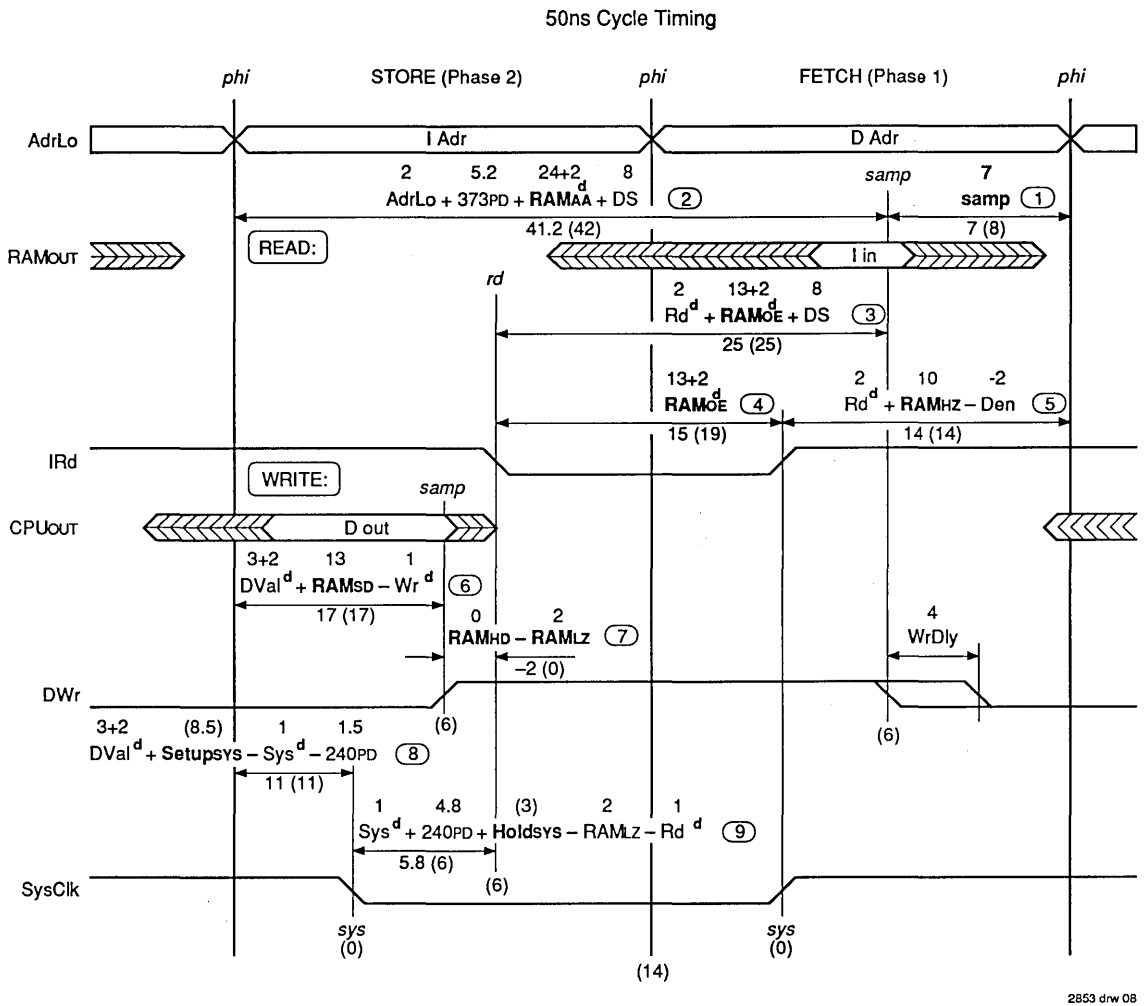
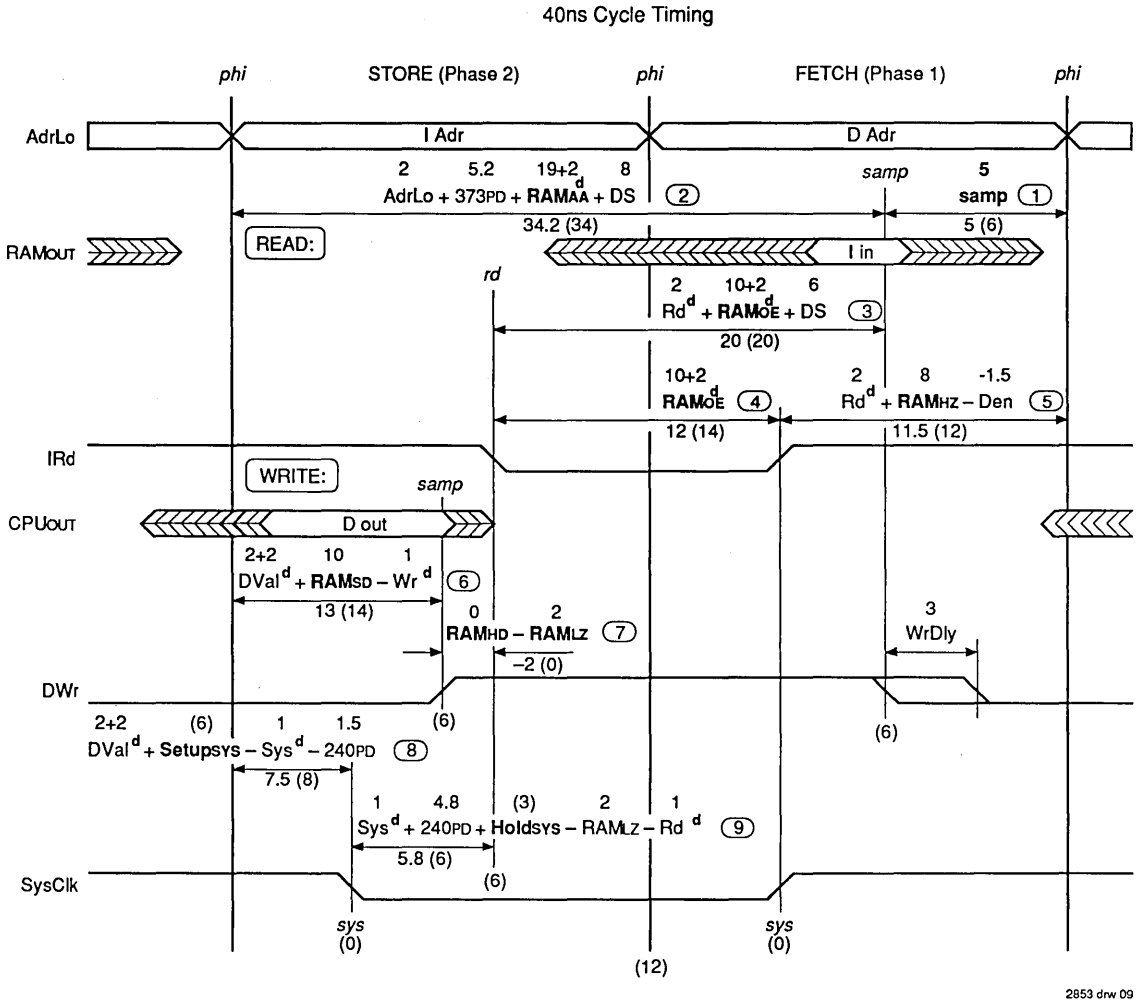


Figure 7. Cache Timing Diagram for a 16MHz IDT79R3000



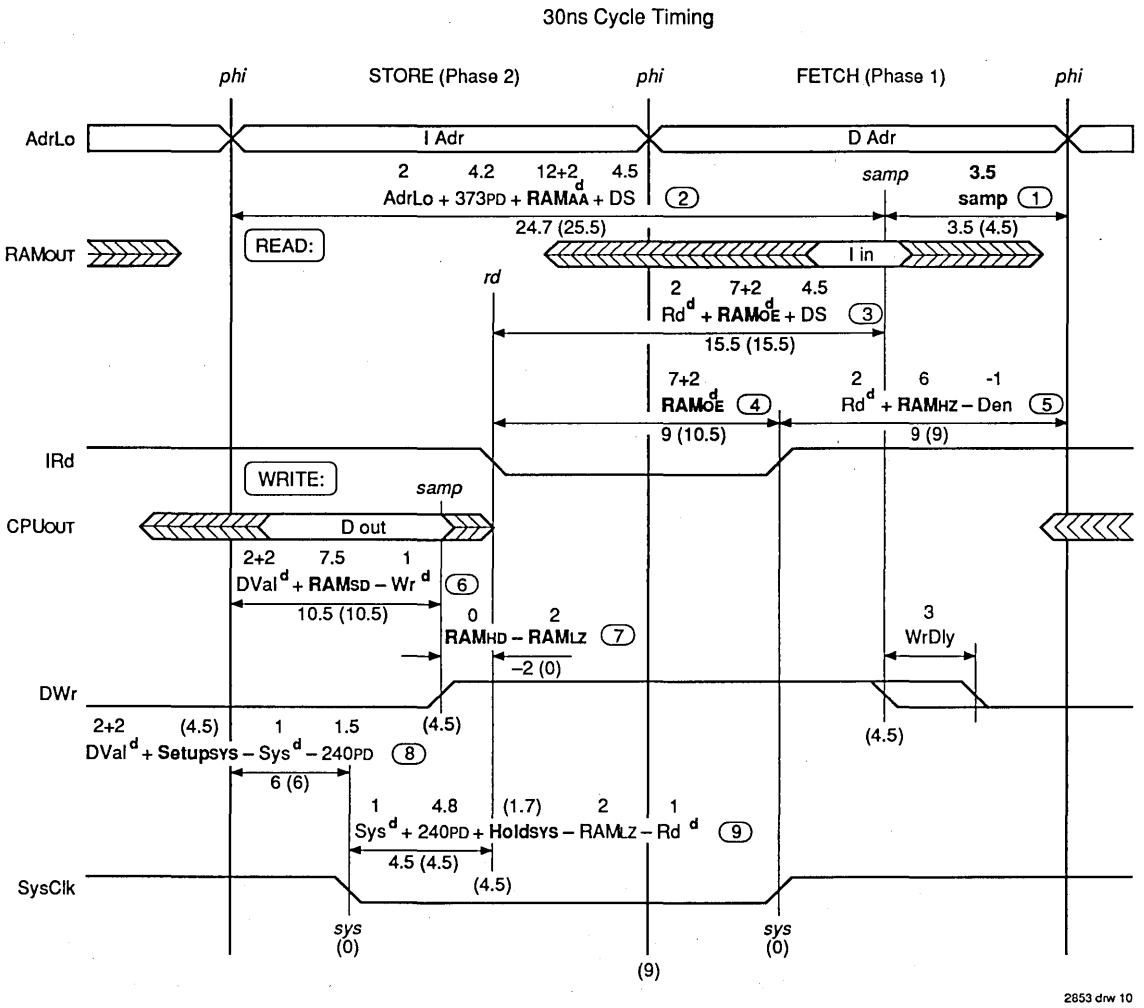
2853 drw 08

Figure 8. Cache Timing Diagram for a 20MHz IDT79R3000



2853 drw 09

Figure 9. Cache Timing Diagram for a 25MHz IDT79R3000



2653 drw 10

Figure 10. Cache Timing Diagram for a 33MHz IDT79R3000

The following equations are used to determine the timing parameters for the static RAM so that they can function as cache for different operating frequencies of the IDT79R3000. The numbers at the left correspond to the encircled numbers in the timing diagrams. Equations 9 and 10 are not shown in the timing diagram but are included for completeness. The equations also use some IDT79R3000 parameters. These are listed in Table 3.

(1) Internal Sample to Phase Delay

This is the time that the processor needs to sample the incoming data. Typically, for the IDT79R3000,

$t_{smp} \geq 5$

(2) RAM Address Access Time

This equation is used to determine the Address Access time parameter requirements of the static RAM. From the timing diagram of Figure 10, it is easily calculated. As an example, let us calculate the address access time for a 33MHz IDT79R3000. The total cycle time for a 33MHz IDT79R3000 is 30ns. If the processor's sample time requirement is met, the time remaining in the cycle is 24ns. In this time the data has to be presented to the processor. The processor requires a data set-up time of 4ns. There is also a propagation delay through the latch for the address bus. For the 33MHz part, a fast IDT54/74FCT373C is used which has a maximum propagation delay of 4.2ns (see Table 4). The derating factors due to the capacitance and the trace length also have to be taken into account. Using all these factors, the equation is:

$t_{RAMAA} \leq t_{cyc} - t_{smp} - t_{DS} - t_{373PD} - t_{AdrLo}^{d} - t_{RAMAA}^{d}$

16MHz IDT79R3000: $t_{RAMAA} \leq 60 - 10 - 9 - 5.2 - 3 - 3$
$t_{RAMAA} \leq 29.8$

20 MHz IDT79R3000: $t_{RAMAA} \leq 50 - 8 - 8 - 5.2 - 2 - 2$
$t_{RAMAA} \leq 24.8$

25 MHz IDT79R3000: $t_{RAMAA} \leq 40 - 6 - 6 - 5.2 - 2 - 2$
$t_{RAMAA} \leq 18.8$

33 MHz IDT79R3000: $t_{RAMAA} \leq 30 - 4.5 - 4.5 - 4.7 - 2 - 2$
$t_{RAMAA} \leq 12.3$

(3) Cache Enable to Sample

This equation is used to determine the system output enable(t_{OES}) requirements of the cache RAM and should meet the processor's set-up specification. The output enable time (t_{OE}) specifications for the RAM are tested for a voltage change of 200mV (a fall from 1.732V to 1.532V for IDT RAMs). For a system, however, the voltage falls from approximately 3.3V to 1.5V. This fall time is usually a nanosecond. **Therefore, the RAM specifications should take this system factor into consideration and specify the output enable time at least one nanosecond lower than the calculated timings.**

$t_{OES} \leq t_{cyc/2} - t_{RD}^{d} - t_{DS} - t_{sys\text{-}smp} + t_{sys\text{-}rd} - t_{OES}^{d}$

16MHz IDT79R3000: $t_{OES} \leq 30 - 3 - 9 - 10 + 10 - 3$
$t_{OES} \leq 15$

20MHz IDT79R3000: $t_{OES} \leq 25 - 2 - 8 - 8 + 8 - 2$
$t_{OES} \leq 13$

25MHz IDT79R3000: $t_{OES} \leq 20 - 2 - 6 - 6 + 6 - 2$
$t_{OES} \leq 10$

33MHz IDT79R3000: $t_{OES} \leq 15 - 2 - 4 - 4.5 + 4.5 - 2$
$t_{OES} \leq 7$

(4) Minimum Read Pulse Width

This timing requirement guarantees that the read pulse width generated by the processor is at least as long as the cache RAM output-enable time.

tOES ≤ tcyc/2 - tsys-rd - tOES^d

16MHz IDT79R3000: tOES ≤ 30 - 10 - 3
tOES ≤ 17

20MHz IDT79R3000: tOES ≤ 25 - 8 - 2
tOES ≤ 15

25MHz IDT79R3000: tOES ≤ 20 - 6 - 2
tOES ≤ 12

33MHz IDT79R3000: tOES ≤ 15 - 4.5 - 2
tOES ≤ 8.5

(5) Read-Write I-Cache Data Bus Contention

This timing requirement ensures that the RAM output is tristated soon enough after the instruction read signal goes high. In the worst case, when the processor performs a store operation, no data contention occurs.

tRAMHZ ≤ tsys - tRd^d + DEn

16MHz IDT79R3000: tRAMHZ ≤ 16 - 3 + (-2.5)
tRAMHZ ≤ 10.5

20MHz IDT79R3000: tRAMHZ ≤ 14 - 2 + (-2)
tRAMHZ ≤ 10

25MHz IDT79R3000: tRAMHZ ≤ 12 - 2 + (-1.5)
tRAMHZ ≤ 8.5

33MHz IDT79R3000: tRAMHZ ≤ 9 - 2 + (-1)
tRAMHZ ≤ 6

(6) Processor Data Set-up to End of Write

This enables the designer to determine whether the cache RAMs have adequate data set-up time when the processor does a store operation. In the equation, the minimum derating is used on the write line i.e., tWr^d, because that is the worst case assumption.

tRAMDS ≤ tcyc/2 - tsys-smp - tDVal - tDVal^d - tWr^d

16MHz IDT79R3000: tRAMDS ≤ 30 - 10 - 3 - 3 - (-2)
tRAMDS ≤ 16

20MHz IDT79R3000: tRAMDS ≤ 25 - 8 - 3 - 2 - (-1)
tRAMDS ≤ 13

25MHz IDT79R3000: tRAMDS ≤ 20 - 6 - 2 - 2 - (-1)
tRAMDS ≤ 11

33MHz IDT79R3000: tRAMDS ≤ 15 - 4.5 - 2 - 2 - (-1)
tRAMDS ≤ 7.5

9

(7) Data Hold from End of Write

This parameter requirement guarantees that the data hold from end of write of the cache RAM is met when the processor or the read buffer is writing to the RAMs.

tRAMHD ≤ tsmp-rd + tRAMLZ

16MHz IDT79R3000: tRAMHD ≤ 0 + 2
tRAMHD ≤ 2

20MHz IDT79R3000: tRAMHD ≤ 0 + 2
tRAMHD ≤ 2

25MHz IDT79R3000: tRAMHD ≤ 0 + 2
tRAMHD ≤ 2

33MHz IDT79R3000: tRAMHD ≤ 0 + 2
tRAMHD ≤ 2

(8) Data SetUp to SysClk

This timing parameter ensures that the set-up time into an external register (for the main memory interface) is sufficient enough for when the processor is doing a store. The data is clocked in the register on the rising edge of the buffered $\overline{\text{SysOut}}$ (through an inverting IDT54/74FCT240A). In this equation, $t_{sys(min)}^{d}$ is used to ensure worst case calculations.

tsetupSys ≤ tcyc/2 - tsys - tDVal - tDVal^d + tsys^d + t240PDmin

16MHz IDT79R3000: tSetUpSys ≤ 30 - 16 - 3 - 3 + 2 + 1.5
tSetUpSys ≤ 11.5

20MHz IDT79R3000: tSetUpSys ≤ 25 - 14 - 3 - 2 + 1 + 1.5
tSetUpSys ≤ 8.5

25MHz IDT79R3000: tSetUpSys ≤ 20 - 12 - 2 - 2 + 1 + 1.5
tSetUpSys ≤ 6.5

33MHz IDT79R3000: tSetUpSys ≤ 15 - 9 - 2 - 2 + 1 + 1.5
tSetUpSys ≤ 4.5

(9) Data Hold from SysClk

This timing parameter is to guarantee that the hold time specification for an external register is met on a processor store. In this equation the minimum value of t_{RD}^{d} is taken to ensure worst case numbers.

tHoldSys ≤ tsys-rd - tsys^d - t240PDmax + tRAMLZ + tRd^d

16MHz IDT79R3000: tHoldSys ≤ 6 - 2 - 4.8 + 2 + 1
tHoldSys ≤ 2.2

20MHz IDT79R3000: tHoldSys ≤ 6 - 1 - 4.8 + 2 + 1
tHoldSys ≤ 3.2

25MHz IDT79R3000: tHoldSys ≤ 6 - 1 - 4.8 + 2 + 1
tHoldSys ≤ 3.2

33MHz IDT79R3000: tHoldSys ≤ 4.5 - 1 - 4.8 + 2 + 1
tHoldSys ≤ 1.9

(10) Address Set-up to End of Write:

This equation enables us to determine the timing requirement for the RAM so that the address set-up time is sufficient before the trailing edge of the write pulse.

$tRAMAW \leq tcyc - tsmp\text{-}sys - tAdrLo^{d} - t373PD + tWr^{d}$

16MHz IDT79R3000: $tRAMAW \leq 60 - 10 - 3 - 5.2 + 3$
$tRAMAW \leq 44.8$

20MHz IDT79R3000: $tRAMAW \leq 50 - 8 - 2 - 5.2 + 2$
$tRAMAW \leq 36.8$

25MHz IDT79R3000: $tRAMAW \leq 40 - 6 - 2 - 5.2 + 2$
$tRAMAW \leq 28.8$

33MHz IDT79R3000: $tRAMAW \leq 30 - 4.5 - 2 - 4.7 + 2$
$tRAMAW \leq 20.8$

(11) Write Hold Pulse-Width:

This requirement guarantees that the cache RAM's minimum write pulse width specification is met.

$tRAMPW \leq tcyc/2 - tWrDly$

16MHz IDT79R3000: $tRAMPW \leq 30 - 5$
$tRAMPW \leq 25$

20MHz IDT79R3000: $tRAMPW \leq 25 - 4$
$tRAMPW \leq 21$

25MHz IDT79R3000: $tRAMPW \leq 20 - 3$
$tRAMPW \leq 17$

33MHz IDT79R3000: $tRAMPW \leq 15 - 2$
$tRAMPW \leq 13$

(12) Write Recovery Time:

The write recovery time is the time between the write pulse going inactive and the change in address. This characteristic is usually specified by the SRAM manufacturer and is typically zero. This parameter is important in the IDT79R3000 cache interface and care must be taken to choose the proper part to prevent race conditions. In the IDT79R3000 cache design using the IDT7198 16 K x 4 RAM, the latch enable is controlled by ICIk/DCIk and the write enable on the RAM is controlled by $\overline{IWr}/\overline{DWr}$. The timing diagram shows the relationship between the two clocks and the parameter t_{WR} Timing calculations below show that the write recovery specifications are not violated.

Derating Calculations for DCIk and $\overline{DWr}$

To calculate the effect of derating on the control signals DCIk and $\overline{DWr}$, the following assumptions have been made:

1) The pin to pin variation on an IDT79R3000 device is 15 % for a 50pF load. Under the maximum case, the deratings will vary from 1.7 to 2ns for DCIk and $\overline{DWr}$. Under the minimum case the deratings will vary from 0.58 to 0.625ns.
2) The trace length for the $\overline{DWr}$ signal is 6 inches.
3) The trace length for the DCIk signal is 2 inches.
4) The trace length of the address bus to the RAM is 4 inches.
5) Each ICIk control signal is connected to four IDT54/74FCT373 devices.

9

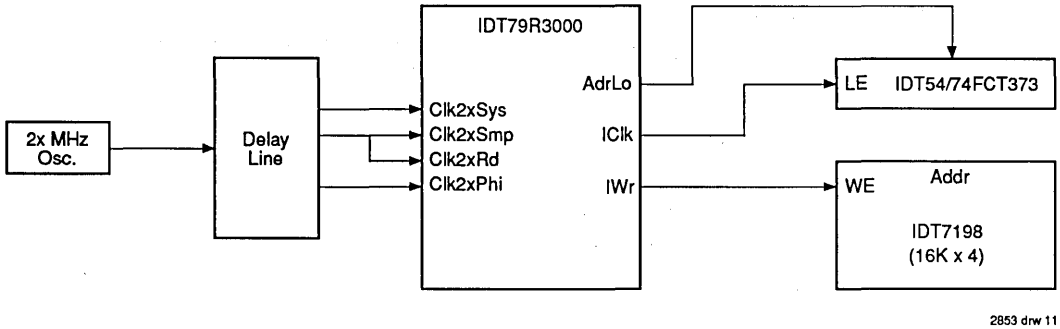


Figure 11. Circuit Showing IWr and ICIk Signals to Latch and SRAM

The input and ouput capacitances for the IDT79R3000, IDT7198 and IDT54/74FCT373 can be obtained from Table 2. Figure 11 is a simple circuit showing the connections of ICIk and $\overline{IWr}$ from the IDT79R3000 to the latch enable (LE) on the IDT54/74FCT373 device and Write Enable ($\overline{WE}$) on the static RAM, respectively. Figure 12 shows the tWR timings with respect to the data cache in an IDT79R3000-based system.

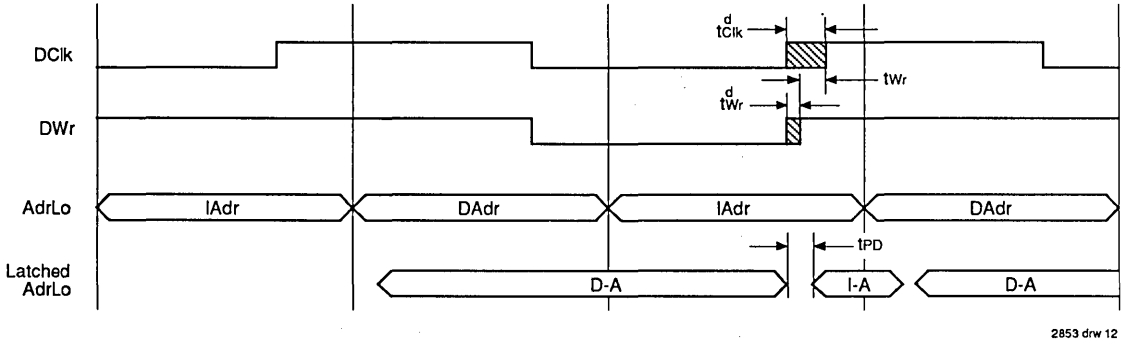


Figure 12. Write Recovery Timing

To prove that the tWR parameter is not violated, the calculations are done as shown below. The derating effects on the DClk and AdrLo signal should exceed that of the $\overline{DWr}$ signal. These calculations are similar to the derating calculations in previous sections. The minimum propagation delay through the latch is considered. The derating on the DClk signal coming out of the IDT79R3000 is less than that of the $\overline{DWr}$ signal. The reverse case is superfluous and, in fact, improves the situation. The minimum and worst case derating effects are shown below. The write recovery time parameter must not be violated over the entire operating range.

Capacitive Derations: (IDT79R3000 Variations 15% 1.7ns - 2ns)

((Cdriver + Cload + Ctrace) - Crated)/25 * (MinOrMax) = CLD

Iclk ((10 + 4 * 10 + 2 * 2) - 25)/25 * **1.7** = 1.97ns

Iwr ((10 + 8 * 7 + 6 * 2) - 25)/25 * **2** = 4.24ns

RamAddr ((12 + 8 * 7 + 4 * 2) - 50)/25 * **0.5** = 0.52ns

Calculations :

Race path 1 : Iclkmin + $t_{PD(Ie)}$ + RamAdrmin = (1.97 + 2 + 0.52) = 4.49
Race path 2 : Iwrmax = 4.24

Path1 - Path2 > t_{WR}

4.49 - 4.24 > 0

Capacitive Derations: (IDT79R3000 Variations 15% 0.58ns - 0.625ns)

((Cdriver + Cload + Ctrace) - Crated)/25 * (MinOrMax) = CLD

Iclk ((10 + 4 * 10 + 2 * 2) - 25)/25 * **0.58** = 0.58ns

Iwr ((10 + 8 * 7 + 6 * 2) - 25)/25 * **0.625** = 1.325ns

RamAddr ((12 + 8 * 7 + 4 * 2) - 50)/25 * **0.5** = 0.52ns

Calculations :

Race path 1 : Iclkmin + $t_{PD(Ie)}$ + RamAdrmin = (0.58 + 2 + 0.52) = 3.1ns
Race path 2 : Iwrmax = 1.325

Path1 - Path2 > t_{WR}

3.1 - 1.325 = 1.775 > 0

From the above calculations and the RAM timing Tables 4 and 5, it can be seen that the data set-up to the processor is met. The output enable of the RAM, which is controlled by $\overline{IRd}$, goes high and the RAM output starts to go tri-state. From the figure, the reader may correctly question whether the hold time requirements of the IDT79R3000 are met. They are indeed met by the capacitance on the bus and also because CMOS devices are being used. The technical note entitled **Meeting Bus Hold for the IDT79R3000** gives a more detailed explanation.

Table 4 gives the timing data sheet for a typical SRAM device. The timing parameters correspond to a particular RAM configuration. Other RAM devices may have different timings for some of the parameters; however, there are certain timings that must be met. These critical parameters are listed in Table 5 and the unlisted parameters may vary a bit from device to device.

9

AC ELECTRICAL CHARACTERISTICS — COMMERCIAL TEMPERATURE RANGE

Symbol	Parameter	16MHz		20MHz		25MHz		33MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
	Clock									
TCkHigh	Input Clock High	12.5	—	10	—	8	—	6	—	ns
TCkLow	Input Clock Low	12.5	—	10	—	8	—	6	—	ns
TCkP	Input Clock Period	30	500	25	500	20	500	15	200	ns
	Clk2xSys to Clk2xSmp	0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xRd	0	tcyc/4	0	tcyc/4	0	tcyc/4	0	tcyc/4	ns
	Clk2xSmp to Clk2xPhi	9	tcyc/4	7	tcyc/4	5	tcyc/4	3.5	tcyc/4	ns
	Run operation									
TDen	Data Enable	—	-2	—	-2	—	-1.5	—	-1	ns
TDDis	Data Disable	—	-1	—	-1	—	-0.5	—	-0.5	ns
TDVal	Data Valid	—	3	—	3	—	2	—	2	ns
TWrDly	Write Delay	—	5	—	4	—	3	—	2	ns
TDS	Data Set-Up	9	—	8	—	6	—	4.5	—	ns
TDH	Data Hold	-2.5	—	-2.5	—	-2.5	—	-1.5	—	ns
TCBS	CpBusy Set-Up	13	—	11	—	9	—	7	—	ns
TCBH	CpBusy Hold	-2.5	—	-2.5	—	-2.5	—	-1.5	—	ns
TAcTy	Access Type [1:0]	—	7	—	6	—	5	—	4	ns
TAT2	Access Type [2]	—	17	—	14	—	12	—	8.5	ns
TMWr	Memory Write	1	27	1	23	1	18	1	9.5	ns
TExe	Exception	—	7	—	7	—	5	—	3.5	ns
	Stall Operation									
TSAVal	Address Valid	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type Valid	—	27	—	23	—	18	—	10	ns
TMRdI	Memory Read Initiate	1	27	1	23	1	18	1	10	ns
TMRdT	Memory Read Terminate	—	7	—	7	—	5	—	3.5	ns
TSd	Run Terminate	2	17	2	15	2	11	2	8	ns
TRun	Run Initiate	—	7	—	6	—	4	—	3	ns
TSMWr	Memory Write	1	27	1	23	1	18	1	9.5	ns
TSEx	Exception Valid	—	20	—	18	—	15	—	10	ns
	Reset Initialization									
TRST	Reset Pulse Width	6	—	6	—	6	—	6	—	TckP
TrstPLL	Reset Timing, PLL[1] On	3000	—	3000	—	3000	—	3000	—	TckP
Trstcp	Reset Timing, PLL Off	128	—	128	—	128	—	128	—	TckP
	Capacitive Load Derating Factor									
CLD	Load Derate	0.5	—	0.5	1	0.5	1	0.5	1	ns/pF

2853 tbl 03

NOTE:
1. PLL: Phase Locked Loops.

Table 3. IDT79R300 AC Specifications

READ CYCLE TIMING SPECIFICATIONS

Parameter	16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tRC	30	—	25	—	20	—	12	—
tAA	—	30	—	25	—	19	—	12(1)
tACS1	—	30	—	25	—	19	—	15
tCLZ1	5	—	5	—	5	—	2	—
tOE	—	15	—	13	—	10	—	7
tOLZ	5	—	5	—	5	—	3	—
tCHZ1	—	12	—	10	—	8	—	8
tOHZ	—	10	—	10	—	8	—	6
tOH	5	—	5	—	5	—	0	—
tPU	0	—	0	—	0	—	0	—
tPD	—	30	—	25	—	20	—	15

2853 tbl 04

WRITE CYCLE TIMING SPECIFICATIONS

Parameter	16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tWC	30	—	25	—	20	—	15	—
tCW1	25	—	21	—	17	—	15	—
tAW	25	—	21	—	17	—	15	—
tAS	0	—	0	—	0	—	0	—
tWP	25	—	21	—	17	—	13	—
tWR1	0	—	0	—	0	—	0	—
tWR2	0	—	0	—	0	—	0	—
tWHZ	—	18	—	16	—	8	—	6
tDW	16	—	13	—	11	—	7	—
tDH	0	—	0	—	0	—	0	—
tOW	5	—	5	—	5	—	5	—

2853 tbl 05

NOTE:
1. This assumes that an IDT54/74FCT373C with a t_{PD} = 4.2ns is used.

Table 4. Static RAM Read and Write Timings to Work as Cache With the IDT79R300

9

READ CYCLE TIMING SPECIFICATIONS[1]

Parameter	16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tRC	*	—	*	—	*	—	*	—
tAA	—	29.8	—	24.8	—	18.8	—	12.8
tACS1	—	*	—	*	—	*	—	*
tCLZ1	*	—	*	—	*	—	*	—
tOE	—	15	—	13	—	10	—	7
tOLZ	*	—	*	—	*	—	*	—
tCHZ1	—	*	—	*	—	*	—	*
tOHZ	—	10.5	—	10	—	8.5	—	6
tOH	*	—	*	—	*	—	*	—
tPU	*	—	*	—	*	—	*	—
tPD	—	*	—	*	—	*	—	*

2853 tbl 06

WRITE CYCLE TIMING SPECIFICATIONS[1]

Parameter	16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tWC	*	—	*	—	*	—	*	—
tCW1	*	—	*	—	*	—	*	—
tAW	44.8	—	36.8	—	28.8	—	19.8	—
tAS	*	—	*	—	*	—	*	—
tWP	25	—	21	—	17	—	13	—
tWR1	*	—	*	—	*	—	*	—
tWR2	*	—	*	—	*	—	*	—
tWHZ	—	*	—	*	—	*	—	*
tDW	16	—	13	—	11	—	7	—
tDH	*	—	*	—	*	—	*	—
tOW	*	—	*	—	*	—	*	—

2853 tbl 07

NOTE:
1. Shown are the minimum or maximum parameters. Numbers not shown are not critical for the IDT79R3000 application.

Table 5. Static RAM Parameters to Work as Cache with the IDT79R3000

Parameter	Load	Symbol	Min.	Max.
IDT54/74FCT373A Propagation Delay	50	T_{373PD}	—	5.2
IDT54/74FCT373A Latch Enable Delay	50	T_{373LE}	2	8.5
IDT54/74FCT373A Latch Enable Hold	50	T_{373Hld}	1.8	—
IDT54/74FCT240A Propagation Delay	50	T_{240PD}	1.5	4.8
IDT54/74FCT373C Propagation Delay	50	T_{373PD}	1.5	4.7
IDT54/74FCT240C Propagation Delay	50	T_{240PD}	1.5	3.7

2853 tbl 08

Table 6. Timing Parameters of IDT54/74FCT Logic Devices

Legend:

tRAMAA - RAM Access Time
tRAMOE - RAM Output Enable Time
tRAMHZ - RAM OutPut Low Impedance to Output in High Impedance
tRAMLZ - RAM Output in High Impedance to Output in Low Impedance
tRAMHD - RAM Data Hold Time

tDS - IDT79R3000 Data Set-up Time
tsys - Phase Difference between Clk2xSys and Clk2xPhi
trd - Phase Difference between Clk2xPhi and Clk2xRd
tsmp - Phase Difference between Clk2xPhi and Clk2xSmp
tcyc - Cycle time of the IDT79R3000
tsmp-rd = tsmp - trd

t240PD - Propagation delay from Clk to Output of IDT54/74FCT240A

9

USING X16 LATCHED RAMS AS CACHE FOR THE IDT79R3000 ON A SURFACE MOUNT DESIGN

Assumptions for Surface Mount Design Layout Using x16 Latched RAMs as Cache for the IDT79R3000

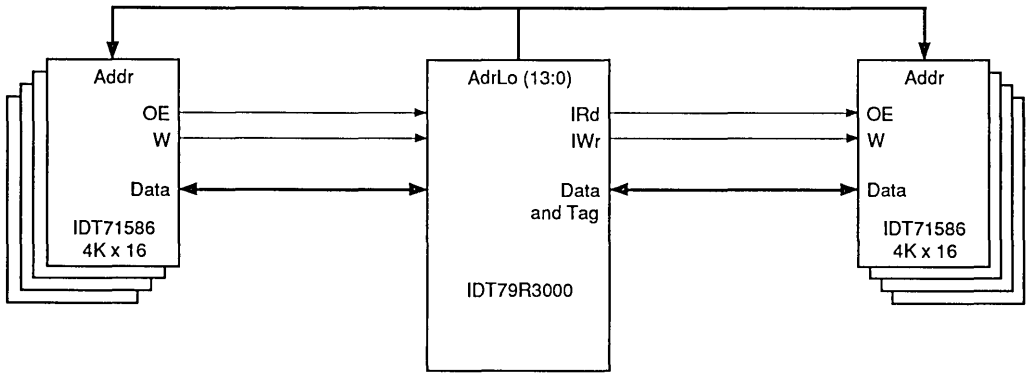
In this section the RAM timings are calculated for a 4K x 16 IDT71586 which has the latches built in. For the static RAMs with latches built in, the address access times, t_{RAMAA}, and the address set-up to end of write, t_{RAMAW} will change from those of a regular static RAM. The propagation delay due to the latches is eliminated, increasing the access time and the address set-up to end of write by about 5ns. In addition, the board layout is different because the distances from the CPU to the RAM are reduced. This decreases the derating factors by a finite amount. This section calculates the derating factors for an IDT79R3000 cache design using the IDT71586 as cache. These are the following assumptions:

1) The trace has a capacitance of 2pF/inch.
2) The speed of light is 2ns/foot in epoxy.
3) The rated load of the IDT79R3000 at which the timings are specified is 25pF. There is a 1ns derating for every additional 25pF.
4) The distances between the IDT79R3000 and the latches are approximately 5 inches.
5) The distances between the IDT79R3000 and the RAMs are approximately 2 inches each.
6) It is assumed that surface mount packages are used. The input capacitance of the RAMs is a typical value (7pF) for a PLCC package.

Derating Calculations Using IDT71586 as Cache RAMs

The derating factors for the IDT71586 cache RAMs follow the same methodology as explained in the section entitled **Dearting Calculations and Cache Timing Considerations Using x4 SRAMs.** The cache size is 4K words for instruction and 4K words for data. The latches are eliminated. The derating factors for the address and data bus are calculated.

Figure 13 shows a cache system for the IDT79R3000 with the latched RAMs (i.e., IDT71586 as the cache). There are a total of 8 such devices required for a 4K word size of instruction and data cache.



For each cache, i.e., Instruction or Data Cache — Size is 4KB
Number of IDT71586s for Data and Tag = 8

2853 drw 13

Figure 13. IDT71586 Used as Cache RAMs for the IDT79R3000, Cache Size = 4KWords

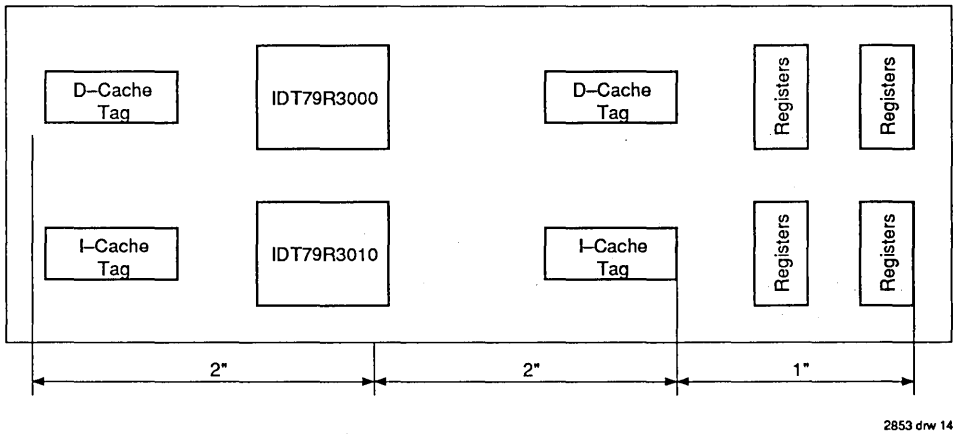


Figure 14. Surface Mount Board Layout of an IDT79R3000 System Using IDT71586 as Cache and Approximate Distances Between Devices

Figure 14 shows an example layout of an IDT79R3000 surface mount design board using latched RAMs (IDT71586) as cache for the IDT79R3000 system. The distance between the IDT79R3000 data pins and the caches is about 2 inches. The total trace length for the address bus and the data bus is about 4 inches each.

Address Bus Derating Calculations

Each AdrLo bus is connected to eight latched RAMs (i.e., the IDT71586) and the address latch for main memory writes and reads. (Figure 15).

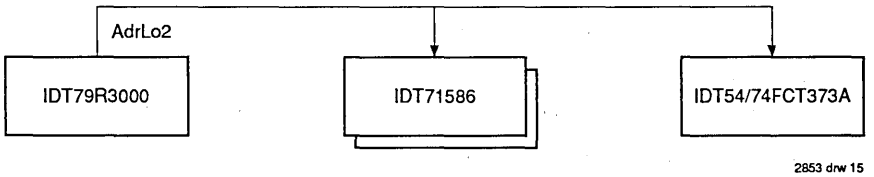


Figure 15. Number of Devices Connected to Address Bus

Trace length from the CPU to the address latch (for main memory) = 4 inches — (74)

Capacitance of the trace = C_{trace} = 4 x 2pF/inch = 8pF — (75)

Input capacitance of the 373 latch = 10pF — (76)

Total input capacitance due to 8 devices = 8 x 7 = 56pF — (77)

Total capacitance due to the load = 8 + 56 + 10 = 74pF — (78)

The rated IDT79R3000 load = 25pF — (79)

Extra loading on the IDT79R3000 = 74 - 25 = 49pF — (80)

The delay can be calculated as follows:

For every extra 25pF of load, there is a delay of 1ns — (81)

From Eq. 80 and Eq. 81, delay due to the capacitive load = 49/25 ≈ 2ns — (82)

9

The speed of light ≈ 2ns/foot —- (83)

For a maximum path length of 3", delay = 3"/12" x 2 = 0.5ns —- (84)

From Eq. 82 and Eq. 84,
Total propagation delay for the address bus = 2 + 0.5 ≈ 2.5ns — (85)

From the above calculations, it is seen that the derating on the address bus is 2.5ns.

Data Bus Derating Calculations

From Figure 16, it is seen that the data bus is connected to the floating point unit (IDT79R3010), two IDT71586 devices, one read register (IDT54/74FCT374A) and one write register (IDT54/74FCT823B). As in the previous section where we considered a 16K x 4 static RAM, we have to calculate the deratings for two cases: i) for an instruction fetch, and ii) for a data store.

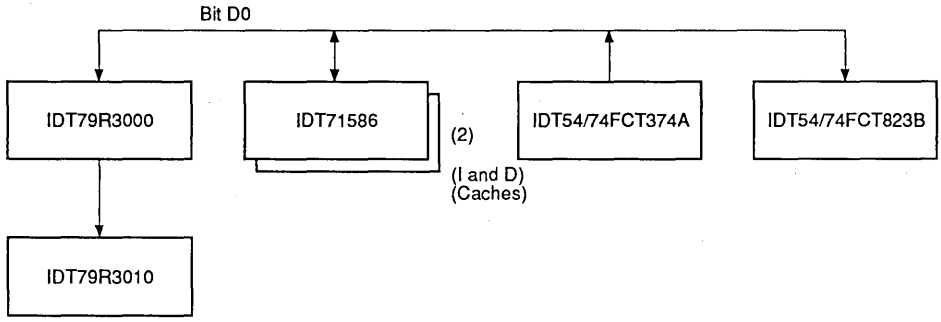


Figure 16. Devices Data Bus Connected to the IDT79R3000

Data Store (IDT79R3000 Outputs Data)

Each data bit is connected to two RAM devices — one for instruction and one for data.

The path length of the data bus = 4 inches — (86)

Trace capacitance of the data bus = 4 x 2pF/inch = 8pF — (87)

Capacitive loading on the data bus due to the different devices = 2 x C_{RAMin} + $C_{R3010in}$ + C_{374out} + C_{823in} = 2 x 7 + 10 + 12 + 10 = 46pF — (88)

Total capacitive load = $C_{devices}$ + C_{trace} = 46 + 8 = 54pF — (89)

Propagation delay due to speed of light = 4"/12" x 2 = 0.6ns —(90)

Delay due to capacitive load = (54 - 25) /25 = 1.16ns — (91)

Total delay = 1.16 + 0.7 = 1.8ns ≈ 2ns— (92)

Load Data Into IDT79R3000 (RAM Outputs Data)

Since the trace length is the same, the trace capacitance C_{trace} = 8pF— (93)

Capacitive load = $C_{R3000in}$ + $C_{R3010in}$ + C_{7}1586in + C_{374in} + C_{823out} — (94)

$C_{devices}$ = 10 + 10 + 7 + 12 + 10 = 49pF — (95)

C_{total} = 49 + 8 = 57pF — (96)

The RAM rated drive = 30pF — (97)

Propagation delay due to extra capacitive loading = (57 - 30) / 25 = 1.08ns — (98)

Propagation delay due to path length = 0.8ns — (99)

Total propagation delay = 1.08 + 0.8 ≈ 2ns — (100)

Read and Write Control Derating Calculations

The effect of the capacitance on the control signals from the IDT79R3000 processor to the caches and the memory interface is considered here. The control signals on the IDT79R3000 are the $\overline{IRd}$, $\overline{DRd}$, $\overline{IWr}$ and $\overline{DWr}$ which control the instruction cache read, data cache read, instruction cache write and data cache write, respectively. The read and write control signals are connected to the output enable ($\overline{OE}$) and write enable ($\overline{WE}$) of the instruction and data cache, respectively. Two control signals each are provided for the read and write operations of each of the caches. Assuming the use of a 4K x 16 IDT71586 static RAM, each control signal is connected to 2 such static RAMs.

Number of devices (SRAM) connected to each control line = 2 — (101)

Input capacitance of each device (SRAM) = 5pF — (102)

Total load capacitance = 2 x 5 = 10pF — (103)

Path length = 4" — (104)

Trace Capacitance = 4 x 2pF/in = 8pF = 8pF — (105)

Total capacitance = 10 + 8 = 18pF — (106)

There is no extra capacitive loading here as the rated IDT79R3000 load is 25pF

Propagation delay due to the trace length = 0.8ns — (107)

Total propagation delay = 0.8 ≈ 1ns — (108)

Timing Equations for Cache Design

This section contains the timing equations that enable us to determine the critical timing requirements of the static RAM that will be used as cache. These equations are based on the use of static RAMs without built-in latches as cache RAMs. The superscript 'd' in the following equations denotes the deratings to be taken into account. The static RAM chosen for illustration is a 4K x 16 IDT71586. The board is assumed to be surface mount for all speeds of the IDT79R3000. The deratings for the surface mount board are 2ns. The deratings were derived from certain assumptions. The explanation and the methodology used is explained in the previous sections. Following is a generalized equation given by the timing requirements for different frequencies of the IDT79R3000. All calculations are based on the IDT79R3000 specifications for the four speed versions (16, 20, 25 and 33MHz), which are found in the IDT data sheets.

(1) Internal Sample to Phase Delay

This is the time that the processor needs to sample the incoming data. Typically, for the IDT79R3000, $t_{smp} \geq 5$.

9

(2) RAM Address Access Time

This equation is used to determine the Address Access time parameter requirements of the static RAM. From the timing diagram of Figure 16, it is easily calculated. The total cycle time for a 33MHz IDT79R3000 is 30ns. If the processor's sample time requirement is met, the time remaining in the cycle is 24ns in which the data has to be presented to the processor. The processor requires a data set-up time of 4ns. The derating factors due to the capacitance and the trace length also have to be taken into account. Using all these factors, the equation is,

$t_{RAMAA} \leq t_{cyc} - t_{smp} - t_{DS} - t_{AdrLo}^{d} - t_{RAMAA}^{d}$

16MHz IDT79R3000: $t_{RAMAA} \leq 60 - 10 - 9 - 3.5 - 3$
$t_{RAMAA} \leq 34.5$

20MHz IDT79R3000: $t_{RAMAA} \leq 50 - 8 - 8 - 2.5 - 2$
$t_{RAMAA} \leq 29.5$

25MHz IDT79R3000: $t_{RAMAA} \leq 40 - 6 - 6 - 2.5 - 2$
$t_{RAMAA} \leq 23.5$

33MHz IDT79R3000: $t_{RAMAA} \leq 30 - 4.5 - 4.5 - 2.5 - 2$
$t_{RAMAA} \leq 16.5$

(3) Cache Enable to Sample

This equation is used to determine the output enable requirements of the cache RAM and should meet the processor's set-up specification. The output enable time for the latched RAM is specified by the manufacturer and tested for a voltage change of 200mV (1.732V to 1.532V for IDT RAMs). For a system, the voltage falls from a level of 3.3V to 1.732 and the added fall time must be considered when specifying the RAM t_{OE} parameter. This fall time is approximately one additional nanosecond. Therefore, the RAM t_{OE} parameter should be one nanosecond lower than the calculated numbers below.

$t_{OES} \leq t_{cyc/2} - t_{RD}^{d} - t_{DS} - t_{sys\text{-}smp} + t_{sys\text{-}rd} - t_{OES}^{d}$

16MHz IDT79R3000: $t_{OES} \leq 30 - 2 - 9 - 10 + 10 - 3$
$t_{OES} \leq 16$

20MHz IDT79R3000: $t_{OES} \leq 25 - 1 - 8 - 8 + 8 - 2$
$t_{OES} \leq 14$

25MHz IDT79R3000: $t_{OES} \leq 20 - 1 - 6 - 6 + 6 - 2$
$t_{OES} \leq 11$

33MHz IDT79R3000: $t_{OES} \leq 15 - 1 - 4 - 4.5 + 4.5 - 2$
$t_{OES} \leq 8$

(4) Minimum Read Pulse Width

This timing requirement guarantees that the read pulse width generated by the processor is at least as long as the cache RAM output enable time.

$t_{OES} \leq t_{cyc/2} - t_{sys\text{-}rd} - t_{OES}^{d}$

16MHz IDT79R3000: $t_{OES} \leq 30 - 10 - 3$
$t_{OES} \leq 17$

20MHz IDT79R3000: $t_{OES} \leq 25 - 8 - 2$
$t_{OES} \leq 15$

25MHz IDT79R3000: $t_{OES} \leq 20 - 6 - 2$
$t_{OES} \leq 12$

33MHz IDT79R3000: $t_{OES} \leq 15 - 4.5 - 2$
$t_{OES} \leq 8.5$

(5) Read-Write I-Cache Data Bus Contention

This timing requirement ensures that the RAM output is tri-stated soon enough after the instruction read signal goes high. In the worst case, when the processor performs a store operation, no data contention occurs.

$t_{RAMHZ} \leq t_{sys} - t_{Rd}^{d} + DEn$

16MHz IDT79R3000: $t_{RAMHZ} \leq 16 - 2 + (-2.5)$
$t_{RAMHZ} \leq 11.5$

20MHz IDT79R3000: $t_{RAMHZ} \leq 14 - 1 + (-2)$
$t_{RAMHZ} \leq 11$

25MHz IDT79R3000: $t_{RAMHZ} \leq 12 - 1 + (-1.5)$
$t_{RAMHZ} \leq 9.5$

33MHz IDT79R3000: $t_{RAMHZ} \leq 9 - 1 + (-1)$
$t_{RAMHZ} \leq 7$

(6) Processor Data Set-Up to End of Write

This enables the designer to determine whether the cache RAMs have adequate data set-up time when the processor does a store operation. In the equation, the minimum derating is used on the write line (i.e., t_{Wr}^{d}) as that is the worst case assumption.

$t_{RAMDS} \leq t_{cyc/2} - t_{sys\text{-}smp} - t_{DVal} - t_{DVal}^{d} - t_{Wr}^{d}$

16MHz IDT79R3000: $t_{RAMDS} \leq 30 - 10 - 3 - 3 - (-2)$
$t_{RAMDS} \leq 16$

20MHz IDT79R3000: $t_{RAMDS} \leq 25 - 8 - 3 - 2 - (-1)$
$t_{RAMDS} \leq 13$

25MHz IDT79R3000: $t_{RAMDS} \leq 20 - 6 - 2 - 2 - (-1)$
$t_{RAMDS} \leq 11$

33MHz IDT79R3000: $t_{RAMDS} \leq 15 - 4.5 - 2 - 2 - (-1)$
$t_{RAMDS} \leq 7.5$

(7) Data Hold from End of Write

This parameter requirement guarantees that the data hold from end of write of the cache RAM is met when the processor or the read buffer is writing to the RAMs.

$t_{RAMHD} \leq t_{smp\text{-}rd} + t_{RAMLZ}$

16MHz IDT79R3000: $t_{RAMHD} \leq 0 + 2$
$t_{RAMHD} \leq 2$

20MHz IDT79R3000: $t_{RAMHD} \leq 0 + 2$
$t_{RAMHD} \leq 2$

25MHz IDT79R3000: $t_{RAMHD} \leq 0 + 2$
$t_{RAMHD} \leq 2$

33MHz IDT79R3000: $t_{RAMHD} \leq 0 + 2$
$t_{RAMHD} \leq 2$

9

(8) Data Set-Up to SysClk

This timing parameter ensures that the set-up time into an external register (for the main memory interface) is sufficient enough for the case when the processor is doing a store. The data is clocked in the register on the rising edge of the buffered $\overline{SysOut}$ (through an inverting IDT54/74FCT240A). In this equation, $t_{sys(min)}^{d}$ is used to ensure worst case calculations.

$t_{setupSys} \leq t_{cyc/2} - t_{sys} - t_{DVal} - t_{DVal}^{d} + t_{sys}^{d} + t_{240PDmin}$

16MHz IDT79R3000: $t_{SetUpSys} \leq 30 - 16 - 3 - 3 + 2 + 1.5$
$t_{SetUpSys} \leq 11.5$

20MHz IDT79R3000: $t_{SetUpSys} \leq 25 - 12 - 3 - 2 + 1 + 1.5$
$t_{SetUpSys} \leq 10.5$

25MHz IDT79R3000: $t_{SetUpSys} \leq 20 - 12 - 2 - 2 + 1 + 1.5$
$t_{SetUpSys} \leq 6.5$

33MHz IDT79R3000: $t_{SetUpSys} \leq 15 - 9 - 2 - 2 + 1 + 1.5$
$t_{SetUpSys} \leq 4.5$

(9) Data Hold from SysClk

This timing parameter is to guarantee that the hold time specification for an external register is met on a processor store. In this equation the minimum value of t_{RD}^{d} is taken to ensure worst case numbers.

$t_{HoldSys} \leq t_{sys\text{-}rd} - t_{sys}^{d} - t_{240PDmax} + t_{RAMLZ} + t_{Rd}^{d}$

16MHz IDT79R3000: $t_{HoldSys} \leq 6 - 2 - 4.8 + 2 + 1$
$t_{HoldSys} \leq 2.2$

20MHz IDT79R3000: $t_{HoldSys} \leq 6 - 1 - 4.8 + 2 + 1$
$t_{HoldSys} \leq 3.2$

25MHz IDT79R3000: $t_{HoldSys} \leq 6 - 1 - 4.8 + 2 + 1$
$t_{HoldSys} \leq 3.2$

33MHz IDT79R3000: $t_{HoldSys} \leq 4.5 - 1 - 4.8 + 2 + 1$
$t_{HoldSys} \leq 1.9$

(10) Address Set-Up to End of Write:

This equation enables us to determine the timing requirement for the RAM so that the address set-up time is sufficient before the trailing edge of the write pulse.

$t_{RAMAW} \leq t_{cyc} - t_{smp\text{-}sys} - t_{AdrLo}^{d} + t_{Wr}^{d}$

16MHz IDT79R3000: $t_{RAMAW} \leq 60 - 10 - 3.5 + 2$
$t_{RAMAW} \leq 48.5$

20MHz IDT79R3000: $t_{RAMAW} \leq 50 - 8 - 2.5 + 1$
$t_{RAMAW} \leq 40.5$

25MHz IDT79R3000: $t_{RAMAW} \leq 40 - 6 - 2.5 + 1$
$t_{RAMAW} \leq 32.5$

33MHz IDT79R3000: $t_{RAMAW} \leq 30 - 4.5 - 2.5 + 1$
$t_{RAMAW} \leq 24$

(11) Write Hold Pulse Width:

This requirement guarantees that the cache RAMs minimum write pulse width specification is met.

$t_{RAMPW} \leq t_{cyc/2} - t_{WrDly}$

16MHz IDT79R3000: $t_{RAMPW} \leq 30 - 5$
$t_{RAMPW} \leq 25$

20MHz IDT79R3000: $t_{RAMPW} \leq 25 - 4$
$t_{RAMPW} \leq 21$

25MHz IDT79R3000: $t_{RAMPW} \leq 20 - 3$
$t_{RAMPW} \leq 17$

33MHz IDT79R3000: $t_{RAMPW} \leq 15 - 2$
$t_{RAMPW} \leq 13$

From the above calculations and Figure 15, it can be seen that the data set-up to the processor is met. The output enable of the RAM, which is controlled by $\overline{IRd}$, goes high and the RAM output starts to go tri-state. From the figure, the reader may correctly question whether the hold time requirements of the IDT79R3000 are met. They are indeed met by the capacitance on the bus and also because CMOS devices are being used. The technical note entitled **Meeting Bus Hold for the IDT79R3000** gives a more detailed explanation.

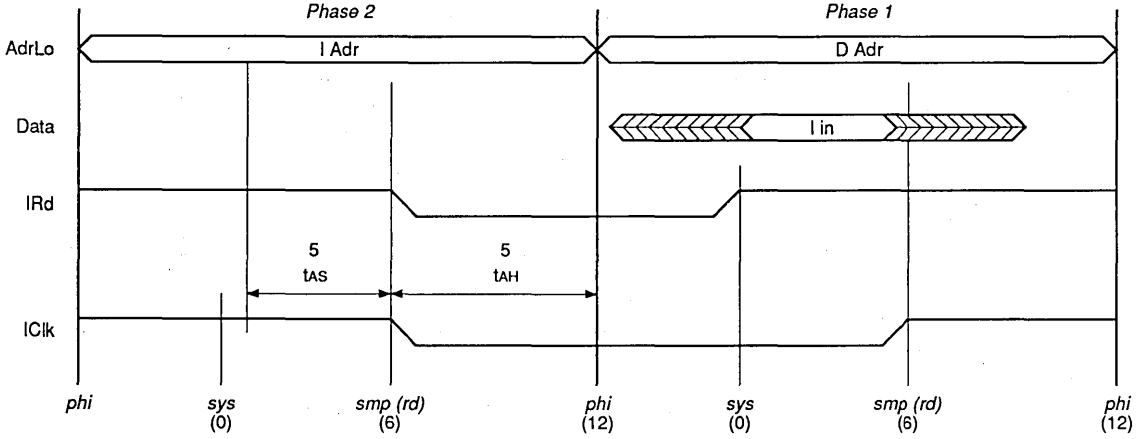


Figure 17. Address Setup and Hold Timing for a Latched RAM (25MHz IDT79R3000)

From Figure 17, it is clearly seen that the address setup and hold time for the latched RAMs are met by using IClk to capture the instruction address. Figure 17 is to illustrate the timings for a 25MHz IDT71586 latched RAM. Similar timing diagrams can be drawn to verify the setup and hold times for IDT79R3000 operating at different frequencies.

9

Legend:

tRAMAA - RAM Access Time
tRAMOE - RAM Output Enable Time
tRAMHZ - RAM OutPut Low Impedance to Output in High Impedance
tRAMLZ - RAM Output in High Impedance to Output in Low Impedance
tRAMHD - RAM Data Hold Time

tDS - IDT79R3000 Data Set-up Time
tsys - Phase Difference between Clk2xSys and Clk2xPhi
trd - Phase Difference between Clk2xPhi and Clk2xRd
tsmp - Phase Difference between Clk2xPhi and Clk2xSmp
tcyc - Cycle time of the IDT79R3000
tsmp-rd = tsmp - trd

t240PD - Propagation delay from Clk to Output of IDT54/74FCT240A

READ CYCLE TIMING SPECIFICATIONS

		16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
Parameter	Description	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tRC	Read Cycle	35	—	30	—	25	—	15	—
tCH	ALEN High	10	—	10	—	10	—	8	—
tCL	ALEN Low	10	—	10	—	10	—	8	—
tAS	Add. Latch Set-Up	5	—	5	—	5	—	4	—
tAH	Add. Latch Hold	5	—	5	—	5	—	4	—
tAA	Address Access	—	35	—	30	—	24	—	15
tACE	Chip Enable Access	—	35	—	30	—	24	—	15
tOE	Output Enable	—	16	—	14	—	11	—	8
tCLZ	CE to Out in LZ	3	—	3	—	3	—	3	—
tOLZ	CE to Out in LZ	2	—	2	—	2	—	2	—
tCHZ	CE to Out in HZ	—	25	—	22	—	20	—	15
tOHZ	CE to Out in HZ	—	11	—	11	—	9	—	7
tOH	Output Hold from Address Change	3	—	3	—	3	—	3	—

2853 tbl 09

Table 7. Read Cycle Timings for an IDT Static RAM with Latches

WRITE CYCLE TIMING SPECIFICATIONS

Parameter	Description	16.7MHZ		20.0MHz		25.0MHz		33.0MHz	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tWC	Write Cycle	35	—	30	—	25	—	15	—
tCH	ALEN High	10	—	10	—	10	—	8	—
tCL	ALEN Low	10	—	10	—	10	—	8	—
tAS	Add. Latch Set-Up	5	—	5	—	5	—	4	—
tAH	Add. Latch Hold	5	—	5	—	5	—	4	—
tAW	Add. to End-of-Write	35	—	30	—	25	—	15	—
tASW	Add. Set-Up	0	—	0	—	0	—	0	—
tWP	Write Pulse Width	25	—	20	—	17	—	11	—
tCW	CE to End-of-Write	25	—	20	—	20	—	11	—
tWR	Write Recovery	0	—	0	—	0	—	0	—
tWHZ	Write to Out in HZ	—	15	—	15	—	13	—	8
tDW	Data Set-Up	16	—	13	—	11	—	7	—
tDH	Data Hold	0	—	0	—	0	—	0	—
tOW	Out Active from End-of-Write	5	—	5	—	5	—	5	—

2853 tbl 10

Table 8. Write Cycle Timings for an IDT Static RAM with Latches

REFERENCES

1) Kane, Gerry, "mips RISC ARCHITECTURE," Prentice-Hall Inc., NJ, 1988

2) IDT RISC R3000 Family Hardware User Manual, October 1988

3) IDT RISC R3000 Family Data Sheets, 1988

4) IDT Data Book, 1988

5) IDT Data Book Supplement, 1989

6) Simha, Satyanarayana , "Meeting Bus Hold for the IDT79R3000 — Technical Note", *Yet to be published*

Integrated Device Technology, Inc.

R3000 FAMILY SOFTWARE TOOLS FOR PERFORMANCE ANALYSIS

APPLICATION NOTE AN–62

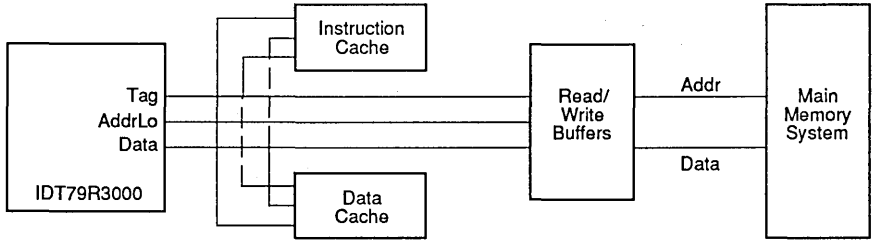
by V.S. Ramaprasad and Roy M. Johnson

INTRODUCTION

System design involves trade-offs between several components to achieve the desired price/performance ratio. Existing development tools of the IDT79R3000 enable the designer to measure and compare the price/performance of various configurations. *cache2000*, Pixie and Pixstats are software tools that allow the designer to analyze the performance of the simulated IDT79R3000 system by executing a designer's application program on the proposed system. Pixie and Pixstats are part of the RISC/os, while *cache2000* is distributed with the System Programmer's Package (SPP). SPP is a set of development tools that include monitors, a standalone C compiler, a standalone I/O library, local and remote debugging tools, downloading software via RS232 and Ethernet, *cache2000* that simulates memory subsystem, and complete system simulation software called SABLE. SABLE simulates IDT79R3000/3001 instructions, caches, TLB behavior, a disk drive and a DUART for a simple console interface. The simulation results of the *cache2000* and Pixstats provide the designer with all the information needed to determine the best possible price/performance solution.

MEMORY SUBSYSTEM

To meet the high processing speed of the MIPS RISC processors, the memory subsystem (Figure 1) is usually structured into a hierarchy of small high-speed cache memory, read/write buffers and large, slow main memory. The cache memory is typically made up of an instruction cache and a data cache. These cache memories allow the CPU to fetch one instruction and one word of data in every clock cycle. To retire the writes to the main memory in one clock cycle, it is common to use a write buffer as an interface between the high-speed CPU and the slower main memory. The write buffer functions as a FIFO of multiple levels and has the ability to perform special functions to minimize the main memory bus traffic. *cache2000* allows the user to choose various parameters for these three basic blocks of the memory hierarchy and then analyze the performance of the proposed memory system.



2854 drw 01

Figure 1. Typical Memory Subsystem IDT79R3000

cache2000

cache2000 is a software tool that simulates a proposed IDT79R3000/3001 memory subsystem. It analyzes the memory references made by an application program during its execution and generates various statistics about its dynamic behavior. *cache2000* determines the execution time taken by the user's application program by simulating the penalties involved in accessing the memory. But, it does not take into account any interlock cycles of the CPU or the FPA. These interlock cycles can be determined from the output of Pixstats. The main memory model simulated is Page mode, and the latencies associated are changeable. By simulating different memory subsystems with the *cache2000*, the user can determine the performance of the application program on those systems and can arrive at an optimal solution.

9

PIXIE

Pixie is a Unix program that generates a new executable of the user's code by adding instructions that trace pieces of code known as basic blocks. This new executable is referred to as pixified executable. A basic block is a sequence of instructions with a single entry point and a single exit point. Once processing starts at the entry point, it will proceed until the exit point without branching. Along with the modified executable, Pixie will also generate a file (with suffix Addrs) that contains the addresses of these basic blocks. When the pixified executable is run (by typing its name), a new file with the execution count of each basic block (with suffix Counts) is generated. Programs like *cache2000* and Pixstats use the Addrs and Counts files, along with the pixified executable, to generate formatted profile information. When using *cache2000*, the user's program must be pixified with -idtrace option to generate memory referencing information of instructions and data. When using Pixstats, the user's program need not be pixified with any option.

PIXSTATS

Pixstats is also a Unix program that analyzes the program's execution characteristics. It generates statistics regarding the opcode frequencies, interlocks and a mini-profile of the program execution. To run Pixstats, the user's program needs to be pixified (running Pixie) without any option. To generate the Counts file, the pixified program must be executed. Pixstats assumes that the user's program fits into the caches and, therefore, does not add any memory penalties. The integer multiply/divide interlock cycles and the floating point interlocks listed in the output of Pixstats needs to be added to the number of cycles given by *cache2000* to accurately determine the total number of cycles the user's program takes for execution. This execution time is with an FPA in the system, as the IDT79R3000/3001 compilers assume the presence of an FPA in the system at the time of code generation. If FPA does not exist, the FPA instructions are emulated in software. Every floating point instruction will cause an exception and the exception handler will invoke the emulating software. A listing of the emulation overhead is provided in the following pages. The overhead and the number of the floating point accelerator instructions contained in the Pixstats output will help the designer in estimating the overhead of a system without an FPA.

MODELING MEMORY SUBSYSTEMS WITH CACHE2000

To simulate different memory subsystems, *cache2000* is modified to the desired parameters of the proposed system. Creating a *cache2000* model to simulate the desired memory subsystem can be done either by editing the source file "cache2000.c", or defining the parameters at Unix command level. All the possible changeable parameters can be assigned new values when the source file is edited. Once the source file is edited, it should be compiled to create the executable of *cache2000*. To compile cache2000.c , an include file called "trace.h" must be in the current directory. Command level modifications can be done at compilation time or at execution time of *cache2000*. The parameters of the caches can be changed by redefining them when invoking the compiler. The main memory latencies and the write buffer depth can be altered at the time of execution. (A complete list of all the options of *cache2000*, Pixie and Pixstats is enclosed at the end of this application note.) The options include:

(a) Editing the source file cache2000.c. — Invoke an editor (vi or emacs) on cache2000.c and modify any changeable parameter. The list of changeable parameters is given at the end. Save the changes and compile the source file with the C compiler. To compile cache2000.c, the file trace.h should also be present in the current directory. The highest level of optimization (O4) is used to produce the most optimal executable. The math library is linked by using -lm. Once cache2000.c is modified with all the required parameter values, there is no need to use any options either at compile time or at run time.

cc -O4 -o cache2000 cache2000.c -lm

(b) Making changes to the Cache2000 at compilation time by redefining the parameters —

cc -O4 -o cache2000 -Di_size_log=12 -Dd_size_log=10 cache2000.c -lm

This compiles a *cache2000* that models an instruction cache of 4K words and data cache of 1K words. The compile time changeable parameters of *cache2000* are:

Parameter Name	Possible Values
i_refill_log	*0,2,4,5*
i_size_log	*10,11,12,13,14,15,16*
d_refill_log	*0,2,4,5*
d_size_log	*10,11,12,13,14,15,16*
byte_gathering	*0,1*
read_conflict_check	*0,1*
istreaming	*0,1*

(c) Setting the parameters at the execution time of *cache2000* — The memory latencies can be set at the time of simulation. *cache2000* is executed in parallel with the pixified executable of the user's application program. The Unix command "makepipe" is used to run these two processes (*cache2000* and user's program) in parallel. The user's program (pixified with -idtrace), when executed, will generate memory referencing information to a Unix file with file descriptor 19. This information is piped (by the makepipe command) to the standard input file (file descriptor 0). *cache2000* reads the standard input and proceeds with the memory subsystem simulation. An

example of setting up the read latency to 6 cycles is:

makepipe 19 my_prog.pixie '|' 0 cache2000 -read_latency 6 > my_prog.cache2000

Now *cache2000* models a main memory with a read memory latency of 6 cycles.

STEPS IN PERFORMANCE ANALYSIS

The steps involved in measuring the performance of the IDT79R3000/3001-based systems are:

(1) Compile the source of the application program with the desired level of optimization to create the executable module.

cc -O4 -o my_prog my_prog.c

(2) Create a *cache2000* executable to model the proposed memory subsystem by compiling the cache2000.c.

cc -O4 -o cache2000 -Di_size_log=12 -Dd_size_log=10 -Di_refill_log= 5 -Distreaming= 1 cache2000.c -lm

This compiles a *cache2000* with instruction cache of 4K words, data cache of 1K words and 32 words of block refill for instruction cache with instruction streaming enabled.

(3) Pixify the application program with *-idtrace.*

pixie -idtrace -o my_prog.pixie my_prog

The above command pixifies the application program *my_prog* and places the output in *my_prog.pixie.*

(4) Use makepipe to run the pixified application program and Cache2000 in parallel.

makepipe 19 my_prog.pixie '|' 0 cache2000 > my_prog.cache2000

This command pipes the memory references generated by *my_prog.pixie* to the *cache2000.* The statistical information produced by *cache2000* to the standard output can be redirected to *my_prog.cache2000.* Arguments, input file, output file for the user's application program can be specified by:

makepipe 19 my_prog.pixie args '<' input_file '>' output_file '|' 0 cache2000 > my_prog.cache2000

By running the makepipe in the background, the designer can fire up more than one simulation. The following command sets up the read latency from the main memory to 6 cycles and runs the simulation in the background.

makepipe 19 my_prog.pixie '|' 0 cache2000 -read_latency 6 > my_prog.cache2000

(5) Run Pixstats to find out the interlock cycles of IDT79R3000/3001 integer multiply/divide unit and the interlock cycles of the FPA IDT79R3010. These interlock cycles should be added to the cycles from the *cache2000* output. The output of Pixstats also contains the IDT79R3000/3001 opcode distribution. This provides the designer with the percentages of the FPA instructions in the application program. Running Pixstats is a three step process. The steps are:

(a) Repixify the application program without any option.
(b) Run the pixified program by typing in the name of the pixified output to generate the basic block counts file.
(c) Run Pixstats.

The commands to accomplish these tasks are:

(d) Pixify the application program.

pixie -o my_prog.pixie myprog

This generates *my_prog.Addrs* file containing the basic block addresses.

(e) Run the pixified program

my_prog.pixie

This generates *my_prog.Counts* file containing the execution count of the basic blocks.

(f) Run Pixstats.

pixstats my_prog > my_prog.pixstats

Pixstats uses the Addrs and Counts files and redirects the output to *my_prog.pixstats.*

cache2000 OUTPUT

The *cache2000* program prints out the statistical information to the standard output file. It can be redirected to any desired file. The statistical information is generated every time after the execution of user-specified number of cycles. By setting the print variable to an extremely large number (20000000000), the output generation can be restricted to the final results.

The following is one such output file for a typical image processing application program written in C language. The executable size is 500Kbytes, the text segment is 300Kbytes and data segment is 200Kbytes consisting mainly of uninitialized data items of size greater than 512 bytes. The *cache2000* models a memory subsystem of 8Kbytes of

9

instruction cache, 8Kbytes of data cache, 4 words of block refill size for the I-cache and 1 word refill for the D-cache. Instruction streaming is enabled. The cache flushes are turned off to model an embedded application. The clock speed is 16.67MHz. The Page mode main memory read latency is 5 cycles, giving an access time of 300ns.

Lines in the file are numbered in the first column for referencing in the explanation.

```
1   cache2000 -flush 20000 -cycle 60 -clock 16.67 -wbsize 1 -read_latency 5:
2   Fri Oct 13 21:39:05 1989

3   73398701 cycles (1.569), 4.4s @ 16.7MHz
4   46783524 instructions (46.8M)
5   0 cache flushes

6   word read       9008972     19.26%
7   double read     0           0.00%
8   word write      4161532     8.90%
9   double write    0           0.00%
10  byte write      445222      0.95%
11  half write      438867      0.94%
12  swr             0           0.00%
13  swl             73          0.00%
14  lwc1            255500      0.55%
15  ldc1            0           0.00%
16  swc1            5635        0.14%
17  sdc1            0           0.00%
18  basic block     8615556     18.42%
19  (null)          0           0.00%

20  I-cache size = 2048 words,direct-mapped, 4 word refill
21  D-cache size = 2048 words,direct-mapped, 1 word refill,write-through
22  Write buffer = 1 deep, conflict checking on d-miss, byte gathering
23  TLBsize = 56 entries, associative, random replacement,
        page size =1024words

                                                per       per       per
                                                instr     cycle     other

24  uTLB misses:            318004      (0.68%/ 0.43%)
25  I-TLB misses:           9394        (0.02%/ 0.01%)
26  D-TLB misses:           7927        (4.29%/ 0.01%/ 0.06%)
27  I-cache misses:         2007056     (0.01%/ 2.73%)
28  D-cache misses:         234278      (0.50%/ 0.32%/ 2.53%)
29  Idle writes:            2171982     (4.64%/ 2.96%/ 42.5%)
                                        (5 memory cycles)
30  Page mode writes:       2561693     (5.48%/ 3.49%/ 50.1%)
                                        (2 memory cycles)
31  Non-page writes:        377581      (0.81%/ 0.51%/ 7.4%)
                                        (6 memory cycles)
32  Total writes:           5111256     (10.93%/ 6.96%)
33  Write merges            0           (0.00%/ 0.00%/ 0.00%)
34  I-stream branch:        351599      (0.75%/ 0.48%/ 17.5%)
35  I-stream d-miss:        31036       (0.07%/ 0.04%/ 1.5%)
36  I-stream write:         252829      (0.54%/ 0.34%/ 12.6%)
37  I-stream block:         1371592     (2.93%/ 1.87%/ 68.3%)
38  I-stream words:         0.4/2.1/1.6
```

```
39  uTLB miss cycles:          318004      ( 0.68%/ 0.43%) (penalty 1)
40  I-TLB miss cycles:         122122      (0.26%/ 0.17%) (penalty 13)
41  D-TLB miss cycles:         103051      (0.22%/ 0.14%) (penalty 13)
42  I-cache miss cycles:       18063504    (38.61%/ 24.61%) (penalty 9)
43  I-cache streaming:         -4155427    (-8.88%/ -5.66%)
44  D-cache miss cycles:       1405668     (3.00%/ 1.92%) (penalty 6)
45  2-cycle SB/SH/SWL/SWR:     884162      (1.89%/ 1.20%) (penalty 1)
46  Write buffer full cycles:  6780214     (14.49%/ 9.24%)
    (average 1.3 per write)
47  Write wait cycles:         3093879     ( 6.61%/ 4.22%)
    (average 1.4 per miss)

48  Subtotal:                  26615177    (56.89%/ 36.26%)
49  Instructions:              46783524    (100.00%/ 63.74%)
50  Total:                     73398701    (156.89%/ 100.00%)
51  TLB memory cycles:         111030      (0.24%/ 0.15%)
52  Memory bus cycles:         33976968    (72.63%/ 46.29%)
53  simulation runtime:        248.4u 23.4s, 5473.6w/5%, 172 Kinst/s, 255 Krefs/s
54  DONE!
```

EXPLANATION OF FIELDS

cache2000 Runtime Parameters

```
1  cache2000 -flush 20000 -cycle 60 -clock 16.67 -wbsize 1 -read_latency 5:
2  Fri Oct 13 21:39:05 1989
```

All the *cache2000* runtime parameters that are set by the user are listed in this line. The cache flush interval is set to 20,000 million cycles. This implies that the caches are flushed for every 20 billion cycles of user's program execution. In Multiprogramming/Multitasking systems, on context-switching, implicit cache flushing occurs due to a series of cache misses. These cache misses are equivalent to caches being flushed. To simulate that environment, *cache2000*allows the user to set the cache flush interval. By choosing a large number (like 20 billion) for this variable, this flushing can be turned off for embedded applications that do not run Multiprogramming/ Multitasking software. The clock speed selected by the user is 16.67MHz and, therefore, the cycle time is 60 nanoseconds. The write buffer simulated is 1 word deep. The Page mode main memory read latency on load misses is 5 cycles or 300 nanoseconds.

Cycles and Instructions

```
3  73398701 cycles (1.569), 4.4s @ 16.7MHz
4  46783524 instructions (46.8M)
```

The user's application program takes 73398701 cycles for complete execution. To execute these cycles, the processor takes 4.4 seconds when running at 16.67MHz. It takes an average of 1.569 cycles per instruction for simulation of the current memory subsystem and of the current user's application program. As we change the parameters for the memory subsystem, notice that this average number of cycles per instruction will also change. For a fine tuned memory subsystem this number approaches one. The clock speed can be set to the desired frequency with -clock option at execution time of the *cache2000*. Line 4 gives the number of instructions executed. The last field in line 4 is the number of instructions in millions. The number of instructions is independent of the memory system being simulated. It only depends on the user's program, the data files the program is using and the optimization level used in compiling the program.

Cache Flushes

```
5  0 cache flushes
```

In the current simulation, the caches are flushed for every 20,000 million instructions to simulate an embedded application environment without any context switching. The cache flushes are zero for this example, as the program does not execute 20 billion instructions.

9

Reads

6	word read	9008972	19.26%
7	double read	0	0.00%

These two lines give the number of read instructions and their percentage of the total number of instructions. In the current simulation, 19.26% of the total instructions are word (32 bits) load instructions. For cached data/instructions, IDT79R3000 treats all load instructions as load word instructions since a word is read from the caches on cache hits. On a cache miss, a word or multiple words are read from the memory. The double reads on line 7 are not applicable to the IDT79R3000.

Writes

8	word write	4161532	8.90%
9	double write	0	0.00%
10	byte write	445222	0.95%
11	half write	438867	0.94%
12	swr	0	0.00%
13	swl	73	0.00%

These lines list all the store instructions executed by the processor. All the load/store instructions work only with the data cache if the load/store address is cached. In this example, almost 9% of the stores are store word instructions. This is almost half of the word read percentage, which is typical of many applications. Double writes are not applicable to the IDT79R3000. For partial word writes (sb,sh,swr,swl), the IDT79R3000 does a read-modify-write operation. The processor does a load from the cache (on cache hit) at the store address, merges the data to be stored with the data loaded and writes the result back to cache. On a cache miss for the load, the partial word is written to the main memory (through a write buffer), leaving the cache untouched.

FPA (cp1) Read and Writes

14	lwc1	255500	0.55%
15	ldc1	0	0.00%
16	swc1	65635	0.14%
17	sdc1	0	0.00%

The number of reads and writes to/from the FPA (coprocessor 1) are given by the above four lines. The double word load and store (ldc1,sdc1) are not applicable to the IDT79R3000. The coprocessor load/store instructions are word operations. The CPU and the coprocessors are tightly coupled as they share the same data bus. The lwc1 and swc1 instructions are executed by the IDT79R3000. For the current user's program, there are 255500 FPA load instructions. The percentage of these loads to the total number of instructions is 0.55. The number of store instructions is 65635 and 0.14 is the percentage to the total number of instructions.

Basic Blocks

18	basic block	8615556	18.42%

A basic block is a sequence of instructions with a single entry point and a single exit point. Once execution starts at the entry point instruction, it proceeds sequentially until the exit point instruction. Entry points are the target addresses of a jump/branch instruction and exit points are addresses holding a jump/branch instruction. The basic block count on line 18 indicates the number of basic blocks of the user's application program executed.

Anulled Instructions

19	(null)	0	0.00%

Not applicable for the IDT79R3000.

Instruction Cache

20 I-cache size = 2048 words,direct-mapped,4 word refill
21 D-cache size = 2048 words,direct-mapped,1 word refill,write-through

The instruction cache and the data cache for the IDT79R3000 are direct-mapped. For stores, the data is written to the main memory through the data cache, implementing the write-through policy. For partial word stores, the data cache is looked up for a cache hit. If it misses, the partial word is written to the main memory leaving the cache unchanged. The I-cache size, D-cache size, I-cache refill size and D-cache refill size can be chosen by the user. In the current simulation, the caches are 2048 words deep (8Kbytes) and the block refill sizes for I-cache and D-cache are 4 words and 1 word, respectively. These parameters can be set at compilation time of the Cache2000. The possible cache sizes are 4, 8, 16, 32, 64 , 128 and 256 Kbytes. The block refill sizes can be selected as 1, 4, 8, 16 or 32 words.

Micro TLB Misses

24 uTLB misses: 318004 (0.68%/ 0.43%)

In translating the virtual address of an instruction to its physical address, a two-entry table called uTLB (micro TLB) is employed. This micro TLB is not used for data references. If a uTLB miss is encountered, the TLB (56 entries) is looked up. Whenever there is a uTLB miss, its least recently used entry is updated with the appropriate entry from the main TLB. The percentage of uTLB misses to the total number of instructions and the percentage of the misses to the total number of cycles are given in paranthesis.

TLB Misses

25 I-TLB misses: 9394 (0.02%/ 0.01%)
26 D-TLB misses: 7927 (0.02%/ 0.01%/ 0.06%)

The above two lines account for the number of times the mapping information needed (a TLB entry) to translate the virtual address to physical is missing from the TLB. The first line gives the number of misses for instructions while the second one accounts for the data references. The main memory holds the entire page table and on a TLB miss, this page table is looked up in order to update the TLB. The TLB is a fully associative memory and a random replacement algorithm is employed for TLB updates from the page table. The last number in paranthesis for D-TLB misses is the percentage of the D-TLB misses to the total number of load and store instructions.

Cache Misses

27 I-cache misses: 2007056 (4.29%/ 2.73%)
28 D-cache misses: 234278 (0.50%/ 0.32%/ 2.53%)

The I-cache misses are the number of times the processor referred to the instruction cache and could not find the instruction. The number of misses on line 27 indicates that the 2Kwords of I-cache are not quite sufficient for 100% hit rate. Increasing the cache size might enhance the performance. The first percentage, 4.29%, is the I-cache miss rate, making the I-cache hit rate 95.71%. The D-cache misses are the number of attempts the processor made to read from the data cache but could not locate the word. The data cache miss rate is given by the last percentage in paranthesis (2.53%) which is the percentage of D-cache misses to the number load word instructions (lw) and the number of load word instructions to the coprocessor 1 (lwc1). The data cache misses, however, do not include the load misses occuring in read-modify-write operations of partial word stores.

Main Memory Writes

29	Idle writes:	2171982	(4.64%/ 2.96%/ 42.5%) (5memory cycles)
30	Page mode writes:	2561693	(5.48%/ 3.49%/ 50.1%) (2 memory cycles)
31	Non-page writes:	377581	(0.81%/ 0.51%/ 7.4%) (6 memory cycles)
32	Total writes:	5111256	(10.93%/ 6.96%)
33	Write Merges	0	(0.00%/ 0.00%/ 0.00%)

cache2000 simulates Page mode main memory. All the different types of writes to the main memory are listed here. Idle writes are writes to the main memory that do not follow or precede another write. Page mode writes are successive writes to the same page. Non-page writes are successive writes to different pages. Total writes are the summation of these three types of writes. The third percentage in the first set of parentheses on lines 29 through 31 is the percentage of the corresponding write to the total writes. The number of memory cycles assumed in the current simulation to perform these writes is given in the second set of parentheses. These cycles can be specified by the user at run time of *cache2000*. Write merges involve merging the words in the write buffer if the words have the same address. These write merges are simulated by Cache2000 if byte gathering is turned on. Byte gathering is one of the features of the write buffer, IDT79R3020. Write merges reduce the number of references made to the main memory for data updates. In this simulation with 1 word deep write buffer and byte gathering turned on, the number of write merges is zero.

Streaming

34	I-stream branch:	351599	(0.75%/ 0.48%/ 17.5%)
35	I-stream d-miss:	31036	(0.07%/ 0.04%/ 1.5%)
36	I-stream write:	252829	(0.54%/ 0.34%/ 12.6%)
37	I-stream block:	1371592	(2.93%/ 1.87%/ 68.3%)
38	I-stream words:	0.4/2.1/1.6	

On every I-cache miss, a block refill number of instructions is brought into the instruction cache. Enabling streaming allows the CPU to execute these instructions as they get written into the instruction cache. In other words, the CPU does not have to wait for the completion of the writing of a block of instructions into the instruction cache before it starts the execution. Line 37 gives the number of times the CPU streamed through a full block. Lines 34 to 36 list the number of times the processor dropped out of streaming due to branches, data cache misses, write buffer flushes and partial word stores. Line 34 gives the number of occasions the processor dropped out of streaming because of executing a branch/jump instruction. If the CPU is executing a load and faces a data cache miss, streaming is aborted; Line 35 accounts for such cases. The processor also drops out of streaming whenever the write buffer is full and the current instruction needs to write to the write buffer, or whenever it has to perform a read-modify-write operation for partial word store instructions. I-stream write on Line 36 gives the number of such drop outs. The last line, 38, gives the instruction number in the block that caused the I-cache miss (0.4 or the second instruction in the block; numbering of the instructions starts from 0 in a block), the number of instructions the processor streamed through (2.1) and the number of instructions of the block the processor did not execute (1.6). The numbers on Line 38 are average numbers. The last percentage in lines 34 to 37, under the other column, is the percentage of the corresponding streaming number to the number of I-cache misses on Line 27. Dropping out of streaming does not stop the refilling of the instruction cache with the block refill number of instructions. When streaming is aborted, the processor waits till the refilling is finished and then starts executing the instruction that caused the drop out.

Micro TLB Miss Cycles

24	uTLB misses:	318004	(0.68%/ 0.43%)
39	uTLB miss cycles:	318004	(0.68%/ 0.43%) (penalty 1)

When the uTLB does not contain the entry to translate the virtual address of an instruction to its physical address, there is a penalty (1 cycle) in referring the main TLB and updating one of its entries with the appropriate entry for future use. The miss cycles in Line 39 are a product of the number of uTLB misses (318004) with the penalty (1) on Line 39. This penalty of 1 cycle can not be changed by the user. The percentages in parentheses on Line 39 are the penalty cycles per instruction and per cycle, respectively.

TLB Miss Cycles

25	I-TLB misses:	9394	(0.02%/ 0.01%)
26	D-TLB misses:	7927	(0.02%/ 0.01%/ 0.06%)
40	I-TLB miss cycles:	122122	(0.26%/ 0.17%) (penalty 13)
41	D-TLB miss cycles:	103051	(0.22%/ 0.14%) (penalty 13)

The penalty cycles for loading the main TLB and/or the uTLB with a single page table entry from the main memory is given by lines 40 and 41. The penalty for every miss (13 cycles) is fixed and can not be set by the user. These miss cycles do not account for any I/D cache miss penalty cycles that might arise while executing the TLB refill algorithm. The percentages in parentheses on lines 40 and 41 are penalty cycles per instruction and per cycle, respectively.

Cache Miss Cycles

27	I-cache misses:	2007056	(4.29%/ 2.73%)
28	D-cache misses:	234278	(0.50%/ 0.32%/ 2.53%)
42	I-cache miss cycles:	18063504	(38.61%/ 24.61%) (penalty 9)
43	I-cache streaming:	-4155427	(-8.88%/ -5.66%)
44	D-cache miss cycles:	1405668	(3.0%/ 1.92%) (penalty 6)

The penalty cycles associated with the I-cache misses and the D-cache misses are given by lines 42 and 44. The penalty for a miss is the sum of read latency of the main memory (user selectable) and the block refill size. In the current simulation the read latency is set to be 5 cycles and the block refill size is chosen to be 4 words for instruction cache and 1 word for the data cache. If instruction streaming is enabled, the penalty for I-cache misses is reduced because of concurrent execution of the instructions with the refill of the instruction cache. This gain in cycles is given in Line 43. The percentages in parentheses on lines 42 to 44 are penalty cycles per instruction and per cycle, respectively.

Partial Word Stores

10	byte write	445222	0.95%
11	half write	438867	0.94%
12	swr	0	0.00%
13	swl	73	0.00%
45	2-cycle SB/SH/SWL/SWR:	884162	(1.89%/ 1.20%) (penalty 1)

Partial word store instructions take two cycles as the processor performs a read-modify-write operation, so the penalty for each store is one cycle. The total penalty cycles due to these instructions is given in Line 45 (884162). The number of 2-cycle stores is the sum of all the partial word stores listed from Line 10 to 13. The percentages in parentheses on Line 45 are the penalty cycles per instruction and per cycle, respectively.

Write Buffer Penalty Cycles

46	Write buffer full cycles:	6780214	(14.49%/ 9.24%) (average 1.3 per write)
47	Write wait cycles:	3093879	(6.61%/ 4.22%) (average 1.4 per miss)

Whenever the write buffer is full, the next write causes the buffer to be flushed. The processor waits while this flush is carried out. For the current user program, the processor waited for 6780214 cycles (Line 46) while the main memory is being updated with the contents of the write buffer. The average wait penalty per write due to a full write buffer is given in second set of parentheses on Line 46 (1.3). The write buffer continually updates the main memory whenever it has some data and the memory bus is free. When there is an I-cache miss or a D-cache miss, the memory bus is busy supplying instructions and data holding off the write buffer writes. The number of cycles the writes are kept waiting due to reads from the main memory on cache misses is given on Line 47. The average number in parentheses is the write wait cycles per I- and D-cache misses.

Subtotal

48 Subtotal: 26615177 (56.89%/ 36.26%)

The subtotal is the total number of penalty cycles minus the cycles gained due to streaming. The percentage of the total penalty cycles to the total instructions and the percentage of the total penalty cycles to the total cycles is given in ( )s.

Instructions

4 46783524 instructions (46.8M)
49 Instructions: 46783524 (100.00%/ 63.74%)

Line 49 gives the total number of instructions of the current application program executed. The number of instructions might change if the application program is data dependent or if a different optimization level is used in compiling the program.

Total Cycles

3 73398701 cycles (1.569), 4.4s @ 16.7MHz
50 Total: 73398701 (156.89%/ 100.00%)

The total number of cycles the processor takes to execute the current user program is the sum of the number of instructions (1 cycle/instr) and the total number of penalty cycles due to uTLB/TLB misses, cache misses, 2-cycle instructions, write buffer flushing and write buffer waits, minus the cycles gained due streaming.

TLB Refill Cycles

51 TLB memory cycles: 111030 (0.24%/ 0.15%)

For every TLB miss, a TLB refill algorithm gets executed. This kernel code is cached and resides in kseg0. The cache misses occuring while this algorithm gets executed are taken into account in I-cache misses and D-cache misses. Line 51 gives the number of cycles spent for such cache misses.

Bus Usage

52 Memory bus cycles: 33976968 (72.63%/ 46.29%)

The caches are loaded with instructions and data from the main memory and updates are made to the main memory by the write buffer keeping the memory bus busy with these transfers. The number of cycles the bus is in usage is given on line 52.

Cache2000 Runtime

53 simulation runtime: 248.4u 23.4s, 5473.6w/5%, 172 Kinst/s, 225 Krefs/s
54 DONE!

The amount of user time in seconds (248.4), the amount of system time in seconds (23.4), the number of instructions executed per second and the number of memory references made per second by the *cache2000* program in simulating the memory model are listed on Line 53. These numbers have no bearing on the performance of the memory model itself.

COMPARISON OF MEMORY SUBSYSTEMS

The application program is run through different memory subsystems and execution times are listed in the following tables. In these simulations, it is assumed that the write buffer is 1 word deep and the main memory read latency is 5 cycles.

	I-Cache Size (Kbytes)	D-Cache Size (Kbytes)	I- & D-Cache Refill (words)	Instruction Streaming	Run Time 16.67MHz (seconds)
(1)	8	0	1	OFF	9.6
(2)	8	0	4	ON	8.3
(3)	8	0	8	ON	8.1
(4)	8	8	1	OFF	5.9
(5)	8	8	4	ON	4.4
(6)	8	8	8	ON	4.2
(7)	16	8	1	OFF	5.1
(8)	16	8	4	ON	4.1
(9)	16	8	8	ON	4.0
(10)	16	6	1	OFF	5.1
(11)	16	16	4	ON	4.1
(12)	16	16	8	ON	4.0
(13)	32	16	1	OFF	4.3
(14)	32	16	4	ON	3.7
(15)	32	16	8	ON	3.7
(16)	32	32	1	OFF	4.2
(17)	32	32	4	ON	3.7
(18)	32	32	8	ON	3.6
(19)	64	32	1	OFF	3.7
(20)	64	32	4	ON	3.5
(21)	64	32	8	ON	3.5
(22)	64	64	1	OFF	3.7
(23)	64	64	4	ON	3.5
(24)	64	64	8	ON	3.5

2854 tbl 01

Table 1.

Notice that when streaming is on, doubling the instruction cache from 8Kbytes to 16Kbytes gives 6% gain in performance, 16Kbytes to 32Kbytes gives 9.5% gain and 32Kbytes to 64Kbytes gives 4.2% gain. When streaming is off, doubling the instruction cache gives an average 16% gain in performance. Also notice that changing the data cache size from 0 to 8Kbytes gives an average gain of 90%.

In the following table, the simulations assume a read latency of 3 cycles and a 1 word deep write buffer. Instruction streaming is turned on.

	I Cache Size (Kbytes)	D Cache Size (Kbytes)	I &D Cache Refill (words)	Run Time 20.00MHz (seconds)
(1)	32	8	1	3.4
(2)	32	8	4	3.0
(3)	32	32	1	3.3
(4)	32	32	4	3.0

2854 tbl 02

Table 2.

Comparing cases 1 and 2, observe that block refill of 4 words enhanced the performance by 13.3%. Between 3 and 4, the gain is 10%. With 1 word refill, incrementing the data cache to 32Kbytes gives a 3% gain. Between 2 and 4, refill size contributes little to the overall performance.

In the following simulations, refill size of both I- and D-caches is 1 word. The read latency is 5 cycles.

	I Cache Size (Kbytes)	D Cache Size (Kbytes)	Write Buffer Depth (words)	Run Time 20.00MHz (seconds)
(1)	4	0	1	9.1
(2)	4	0	4	8.9

2854 tbl 03

Table 3.

Write buffer, for this application program, gives a boost of 2.25% to the performance.

9

PIXSTATS OUTPUT

The following listing is the output of the Pixstats program. The information that is not relevant to performance analysis is omitted from the output file. The interlock cycles of the integer mult/div of the IDT79R3000 and the interlock cycles of the IDT79R3010 add/mult/div units are listed in the Pixstats output. These interlock cycles should be added to the number of cycles given in the *cache2000* output to obtain the exact number of cycles for program execution, since *cache2000* only evaluates the memory penalties and does not take into account of any interlock penalties. On the other hand, Pixstats assumes that the application program fits completely in the caches. Pixstats is run only once because it is independent of the memory subsystem. The number of interlock cycles that should be added to the *cache2000* cycles is the difference between the number of cycles given on the first line and the number of instructions on the second line of Pixstats output.

The Pixstats output file is interspersed with comments for explanation. The output file follows.

pixstats embedded_application:

49268275 (1.053) cycles (2.95s @ 16.67MHz) — This line gives the number cycles the IDT79R3000 and IDT79R3010 take to execute the application program. The number of cycles calculated by Pixstats can differ from the cycles calculated by *cache2000*. In calculating the cycles, Pixstats assumes that the application program instructions and data completely reside in the caches. In other words, Pixstats assumes 100% cache hit rate for both I- and D-caches, while Cache2000 accounts for the memory overhead. On the other hand, pixstats estimates the interlock cycles of the IDT79R3000 and IDT79R3010. The interlock cycles are not taken into account by *cache2000*. The number in the first set of ( )s is the average number of cycles per instruction. The execution time is 2.95s at 16.67MHz.

46783524 (1.000) instructions — This line gives the number of instructions executed. Both Pixstats and *cache2000* execute the same number of instructions.

9264472 (0.198) loads — This line gives the total number of loads to the IDT79R3000 and IDT79R3010. This is the sum of loads on lines 6 and 14 of *cache2000* output.

5111329 (0.109) stores — This line gives the total number of stores from the IDT79R3000 and IDT79R3010. This is the sum of stores on lines 8,10,11,12,13, and 16 of Cache2000 output.

14375801 (0.307) loads+stores — These are the memory referencing instructions executed by the IDT79R3000.

14558230 (0.311) data bus use — The number of cycles the data bus is busy is given here.

6406647 (0.137) branches.

7024133 (0.150) nops — The total number of nops executed is given. These nops are due to load delay slots and branch delay slots.

1974000 (0.042) multiply/divide interlock cycles (12/35 cycles) — The IDT79R3000 has a separate integer multiply/divide unit that takes 12 and 35 cycles for integer multiplication and division, respectively. Any attempt to read the result of a multiply/divide before the operation is complete will cause the CPU to interlock until the operation is finished. The number of cycles the units were interlocked is given above.

385600 (0.008) floating point data interlock cycles.
9562 (0.000) floating point add unit interlock cycles.
16676 (0.000) floating point multiply unit interlock cycles.
372 (0.000) floating point divide unit interlock cycles.
98541 (0.002) other floating point interlock cycles — The above lines give the interlock cycles of the IDT79R3010 FPA.

Data interlocks occur because of data dependencies between various floating point operations. This usually happens because the source operand (register) of some fp operation is the destination register of some previous fp operation that has not completed. The interlock cycles due to data dependencies between two successive floating point operations are 385600.

Cycles in which the add/mult/div are interlocked are because consecutive fp operations could not be issued. The add, multiply and divide units themselves are not pipelined and, therefore, a new operation cannot be started before the previous one completes.

0.337 load nops per load — This number indicates that 67% of the load delay slots are filled with useful instructions.

0.356 stores per memory reference — This number implies that 35% of the memory references are store operations and the remaining 65% are load operations. Many application programs typically have a 2:1 ratio for reads to writes.

0.474 branch nops per branch — 53% of the branch delay slots have been successfully filled with useful instructions by the assembler.

Opcode Distribution:

spec	14871837	31.79%
lw	6261479	13.38%
addiu	4970770	10.63%
sw	4161532	8.90%
lnop	3119580	6.67%
bnop	3039486	6.50%
addu	2227361	4.76%
andi	1938082	4.14%
beqz	1881668	4.02%
bnez	1648279	3.52%
sll	1524644	3.26%
lbu	1315591	2.81%
li	1255554	2.68%
lui	1244693	2.66%
b	1030526	2.20%
sra	866987	1.85%
lhu	822784	1.76%
cop1	759725	1.62%
jr	728865	1.56%
sltu	724597	1.55%
lh	607812	1.30%
jnop	603324	1.29%
bcond	536865	1.15%
bne	519056	1.11%
bgez	492881	1.05%
sb	445222	0.95%
sh	43886	0.94%
sltiu	409184	0.87%
beq	405527	0.87%
srl	389980	0.83%
subu	333942	0.71%
jal	332866	0.71%
slt	298293	0.64%
nop	261743	0.56%
lwc1	255500	0.55%
and	225631	0.48%
blez	202636	0.43%
bgtz	182090	0.39%
fcvtd	175925	0.38%
mflo	129374	0.28%
slti	127245	0.27%
fcvts	118090	0.25%
multu	105875	0.23%
jalr	103051	0.22%
mtf	65771	0.14%
swc1	65635	0.14%
or	60680	0.13%
ori	52753	0.11%
bff	50075	0.11%
fmul	48172	0.10%
bltz	43984	0.09%
cff	40491	0.09%
ctf	40046	0.09%
fadd	38012	0.08%
mff	36121	0.08%
c.lt	35951	0.08%
xor	29882	0.06%
srlv	23727	0.05%
xori	23585	0.05%
div	22679	0.05%
sllv	22030	0.05%
fcvtw	19601	0.04%
c.le	19410	0.04%
fsub	17003	0.04%
addi	14746	0.03%
bft	14744	0.03%
fdiv	13146	0.03%
mfhi	12038	0.03%
fneg	9690	0.02%
c.eq	8581	0.02%
divu	7802	0.02%
fmov	7283	0.02%
sub	4142	0.01%
srav	2869	0.01%
nor	2389	0.01%
lb	964	0.00%
c.olt	877	0.00%
syscall	793	0.00%
fabs	736	0.00%
lwl	178	0.00%
lwr	164	0.00%
swl	73	0.00%
add	73	0.00%
j	36	0.00%

The lines above give the opcode distribution of the instructions that are executed. Notice that 759725 COP1 instructions have been executed. This number and the percentage are cumulative for all the COP1 instructions. Less than 2% of the instructions executed are, therefore, IDT79R3010 instructions.

24424 static instructions
4414 static basic blocks
5.5 static instructions per basic block
Static opcode frequency

spec	6646	27.21%
lw	3408	13.95%
sw	2438	9.98%
addiu	2157	8.83%
addu	1627	6.66%
cop1	1160	4.75%
lnop	1094	4.48%
li	1031	4.22%
jal	917	3.75%
bnop	889	3.64%
lui	810	3.32%
beqz	616	2.52%
lwc1	526	2.15%
bnez	518	2.12%
lbu	514	2.10%

lh	500	2.05%
sll	499	2.04%
jr	474	1.94%
sh	464	1.90%
b	455	1.86%
andi	407	1.67%
jnop	406	1.66%
sb	378	1.55%
sra	324	1.33%
subu	316	1.29%
fcvtd	274	1.12%
swc1	259	1.06%
bne	256	1.05%
sltu	225	0.92%
nop	202	0.83%
fcvts	193	0.79%
beq	189	0.77%
lhu	187	0.77%
slt	74	0.71%
fmul	117	0.48%
mtf	117	0.48%
bcond	105	0.43%
ori	105	0.43%
blez	104	0.43%
fadd	103	0.42%
slti	100	0.41%
srl	88	0.36%
mflo	80	0.33%
bgez	78	0.32%
and	75	0.31%
sltiu	61	0.25%
multu	57	0.23%
bgtz	50	0.20%
mff	49	0.20%
bff	44	0.18%
fdiv	42	0.17%
cff	37	0.15%
ctf	34	0.14%
c.lt	32	0.13%
fsub	29	0.12%
bltz	27	0.11%
jalr	22	0.09%
xori	20	0.08%
div	20	0.08%
xor	19	0.08%
lb	17	0.07%
fcvtw	16	0.07%
c.le	15	0.06%
bft	15	0.06%
addi	14	0.06%
fneg	13	0.05%
fmov	12	0.05%
sllv	9	0.04%
syscall	9	0.04%
or	9	0.04%
c.eq	9	0.04%
divu	7	0.03%
mfhi	6	0.02%
fabs	6	0.02%
srlv	5	0.02%
lwl	4	0.02%
lwr	4	0.02%
srav	4	0.02%
sub	3	0.01%
c.olt	3	0.01%
j	2	0.01%
swl	2	0.01%
add	2	0.01%
nor	1	0.00%

The list above gives the number of different kinds of instructions the compiler has generated. The total number of instructions the compiler generated is 24424 and, in the executable, there are 1160 COP1 instructions. The numbers given in the list are cumulative of their kind.

FPA EMULATION OVERHEAD

The MIPS compilers generate executable code assuming the presence of an FPA. When the FPA is not present, or if it is not functional, for every FPA instruction, the exception handler invokes the floating point instruction emulation software. The current state execution information is passed onto the emulation software through the Process Control Block (pcb). Following the IEEE standards, the emulation software decodes the FPA instructions, fetches the operands, checks for NaNs, Infinities, Zeros and Denormalized numbers, carries out the FPA operation, normalizes the result and, finally, returns the result.

The amount of time it takes to execute the emulation software depends on the instruction being emulated, the value of the operands and the amount of the emulation code in the cache. The following is a table listing of the emulation times of the emulation software of RISC/os 4.0 on M/120 with IDT79R3000 @ 16.67MHz for the floating point arithmetic instructions. The times are in micro seconds. The numbers in ( )s are the number of times the emulation is slower than actual execution on the IDT79R3010.

Instruction	Best Case µsecs	Avg Case µsecs	Worst Case µsecs
ADD.S	8.35(70)	18.31(153)	25.5(213)
ADD.D	8.35(70)	18.31(153	25.5(213)
SUB.S	8.35(70)	18.31(153)	25.5(213)
SUB.D	8.35(70)	18.31(153)	25.5(213)
MUL.S	9.48(40)	23.08(96)	42.91(179)
MUL.D	9.48(32)	23.08(77)	42.91(143)
DIV.S	9.48(13)	23.08(32)	42.91(60)
DIV.D	9.48(9)	23.08(20)	42.91(38)

2854 tbl 04

Table 4.

The best case occurs when most of the emulation software is in the cache and the operands are either zero or infinity. On the other hand, the worst case arises when very little of the emulation code is in the cache and the operands are denormalized. The average times are good indicators of the overhead for regular data. To determine the emulation overhead for an application program, the output of the Pixstats must be used. The static and the dynamic opcode distribution for coprocessor 1 listed in the output and the times given above help the designer estimate the total emulation overhead.

CONCLUSION

The software tools — *cache2000*, Pixie and Pixstats — allow the user to accurately project the performance of different IDT79R3000-based systems. The application program needs to be compiled only once. *cache2000* executable must be created for every proposed memory model. All these models can be run in parallel as background jobs. To find the interlock cycles of the IDT79R3000 and IDT79R3010, Pixstats should be run only once. The *cache2000* output clearly points out the tunable parameters of the memory subsystem. The Pixstats output provides information on integer and floating point interlocks and static/dynamic opcode distribution.

9

APPENDIX A

cache2000(lspp) SYSTEM PROGRAMMER'S MANUAL

Name

cache2000 — analyze cache misses for an M2000 system (IDT79R2900/R2950)

Synopsis

Cache2000 [options]

Description

cache2000 analyzes a memory reference stream produced by a program pixified with –idtrace. To use ***cache2000***, first use **Pixie**(1) to translate and instrument the executable object module for the program. Use pixie's –idtrace option. Next, execute the translation on an appropriate input using **makepipe**(1). The pixified program will output the addresses and types of memory references to Unix file descriptor 19. With **makepipe** this trace can be fed into the standard input of ***cache2000*** for simulation. Example:

```
makepipe 19 foo.pixie fooarg '<' fooinput '>' foooutput
'|' 0 \
  /usr/local/bin/cache2000 –comment "foo fooarg"
  '>' foo.cache2000
```

cache2000 differs from **cache26**(1) in that it models the IDT79R2950 memory board, which supports page mode writes and 16–word cache refill. The IDT79R2900 also uses the IDT79R3000 2–cycle partial word store option to avoid cache invalidation. The cache sizes are hardwired in the source to 64K bytes. The data cache is write-thru, with IDT79R2020 write buffering. ***cache2000*** simulates the memory subsystem. Accurate performance predictions must add the stall cycles predicted by **pixstats**(1).

–comment **string**
: Include **string** in the output. This is useful for associating the output with the program and input used to generate it.

–print **N**
: Set the interval for periodic statistics printout. **N** is the interval in millions of instructions. Default is 20 million instructions.

–flush **N**
: Set the cache flush interval. **N** is the interval in millions of instructions. Default is 1 million instructions.

–[no]random_flush
: Flush cache at random intervals (Poisson distribution). Default: –norandom_flush.

–cycle **ns**
: Use a **ns** cycle time when converting cycle counts to seconds. Default is 40ns.

–clock **MHz**
: Use a 1000/**MHz** cycle time when converting cycle counts to seconds. Default is 25.0MHz.

–wbsize **N**
: Simulate a **N**-deep IDT79R2020 write buffer. Default is 4 deep. Maximum is 8 deep.

–read_latency **N**
: Instruction and data cache misses take **N**+16 extra cycles. Default is 12 cycles.

–idle_word **N**
: Set the IDTR2950 memory board idle word write time to **N** cycles. Default is 4 cycles.

–page_write **N**
: Set the IDTR2950 non-idle, page mode write time to **N** cycles. Default is 2 cycles.

–nonpage_write **N**
: Set the IDTR2950 non-idle, non-page mode write time to **N** cycles. Default is 6 cycles.

See Also

pixie(1), pixstats(1), makepipe(1), cache23(1), cache26(1), **The MIPS System Programmer's Reference.**

These are the *cache2000* parameters that can be modified to model different memory systems. The values assigned to these parameters are just examples.

```
/* cahce parameters */

/* instruction refill size, in words */
#ifndef i_refill_log
#       define i_refill_log 4
#endif
#define i_refill_size (1<<i_refill_log)
/* instruction cache size, in words */
#ifndef i_size_log
#       define i_size_log 14
#endif
#define i_refill_size (1<<i_refill_log)
/* instruction cache size, in words */
#ifndef i_size_log
#       define i_size_log 14
#endif
#define i_size (1<<i_size_log)
```

```
/* data refill size, in words */
#ifndef d_refill_log
#       define d_refill_log 4
#endif
#define d_refill_size (1<<d_refill_log)
/* data cache size, in words */
#ifndef d_size_log
#       define d_size_log 14
#endif
#define d_size (1<<d_size_log)

/* byte gathering */
#ifndef byte_gathering
#       define byte_gathering 0
#endif
/* read conflict checking in write buffer */
#ifndef read_conflict_check
#       define read_conflict_check 0
#endif
/* instruction streaming */
#ifndef istreaming
#       define istreaming 1
#endif

/* memory parameters */
private unsigned wbsize = 4;
private unsigned read_latency = 13;
private unsigned idle_write_time = 4;
private unsigned page_write_time = 2;
private unsigned nonpage_write_time = 6;
private unsigned byte_extra_write_time = 4;

private char *comment = NULL;
private boolean random_flush = false;
private unsigned print_interval = 20000000;
private unsigned flush_interval = 1000000;
private double random_flush_parameter;
private double cycletime = 40e-9;
```

PIXIE(1-SysV)
RISC/os REFERENCE MANUAL

Name
pixie — add profiling code to a program

Synopsis
pixie in_prog_name [options]

Description
Pixie reads an executable program, partitions it into basic blocks, and writes an equivalent program containing additional code that counts the execution of each basic block. (A basic block is a region of the program that can be entered only at the beginning and exited only at the end). Pixie also generates a file containing the address of each of the basic blocks.

When you run the pixie-generated program, it will (provided it terminates normally or via a call to **exit**(2)) generate a file containing the basic block counts. The name of the file is that of the original program with any leading directory names removed and ".Counts" appended. **prof**(1) and **pixstats**(1) can analyze these files and produce a listing of profiling data.

–[no]quiet
: [Permits] or suppresses messages summarizing the binary-to-binary translation process. Default: –noquiet.

–[no]branchcounts
: –branchcounts inserts extra counters to track whether each branch instruction is taken or not taken. When this option is used, **pixstats** will automatically print more statistics. Default: –nobranchcounts.

–[no]idtrace
: [Disable] or enable tracing of instruction and data memory references. –idtrace is equivalent to using both –itrace and –dtrace together. Default: –noidtrace.

–[no]itrace
: [Disable] or enable tracing of instruction memory references. Default: –noitrace.

–[no]dtrace
: [Disable] or enable tracing of data memory references. For the moment, –dtrace requires –itrace. Default: –nodtrace.

–idtrace_file **number**
: Specify a UNIX file descriptor number for the trace output file. Default: 19.

–bbaddrs **name**
: Specify a name for the file of basic block addresses. Default is to remove any leading directory names from the in_prog_name and append ".Addrs".

–bbcounts **name**
: Specifies the full filename of the basic block counts file. Default: objfile.Counts.

–mips1
: Use the MIPS1 instruction set (IDT79R2000, IDT79R3000) for output executable. This is the default.

–mips2
: Use the MIPS2 instruction set (a superset of MIPS1) for output executable.

See Also
prof(1), **pixstats**(1).
The MIPS Languages Programmer's Guide.

Bugs
The handler function address to the signal system calls is not translated, and so programs that receive signals will not work pixified.

Programs that call **vfork**() will not work pixified because te child process will modify the parent state required for pixie operation. Use **fork**() instead.

Pixified code is substantially larger than the original code. Conditional branches that used to fit in the 16-bit branch displacement field may no longer fit, generating a pixie error.

PIXSTATS(1-SysV)
RISC/os REFERENCE MANUAL

Name

pixstats — analyze program execution

Synopsis

pixstats program [options]

Description

Pixstats analyzes a program's execution characteristics. To use **pixstats**, first use **Pixie**(1) to translate and instrument the executable object module for the program. Next, execute the translation on an appropriate input. This produces a **.Count** file. Finally, use **pixstats** to generate a detailed report on opcode frequencies, interlocks, a mini-profile, and more.

–cycle **ns**
Assume a **ns** cycle time when converting cycle counts to seconds.

–r2010
Use r2010 floating point chip operation times and overlap rules. This is the default.

–r2360
Use r2360 floating point board operation times and overlap rules.

–disassemble
Disassemble and show the analyzed object code.

See Also

pixie(1), prof(1), **The MIPS Languages Programmer's Guide.**

Bugs

Pixstats models execution assuming a perfect memory system. Cache misses etc. will increase above the **pixstats** predictions.

Integrated Device Technology, Inc.

USING IDT73200 OR IDT73210 AS READ AND WRITE BUFFERS WITH R3000

APPLICATION NOTE AN-65

CONTENTS

Integrated Device Technology, Inc.

USING THE IDT73200 MULTILEVEL PIPELINE REGISTERS AS READ AND WRITE BUFFERS WITH R3000/1

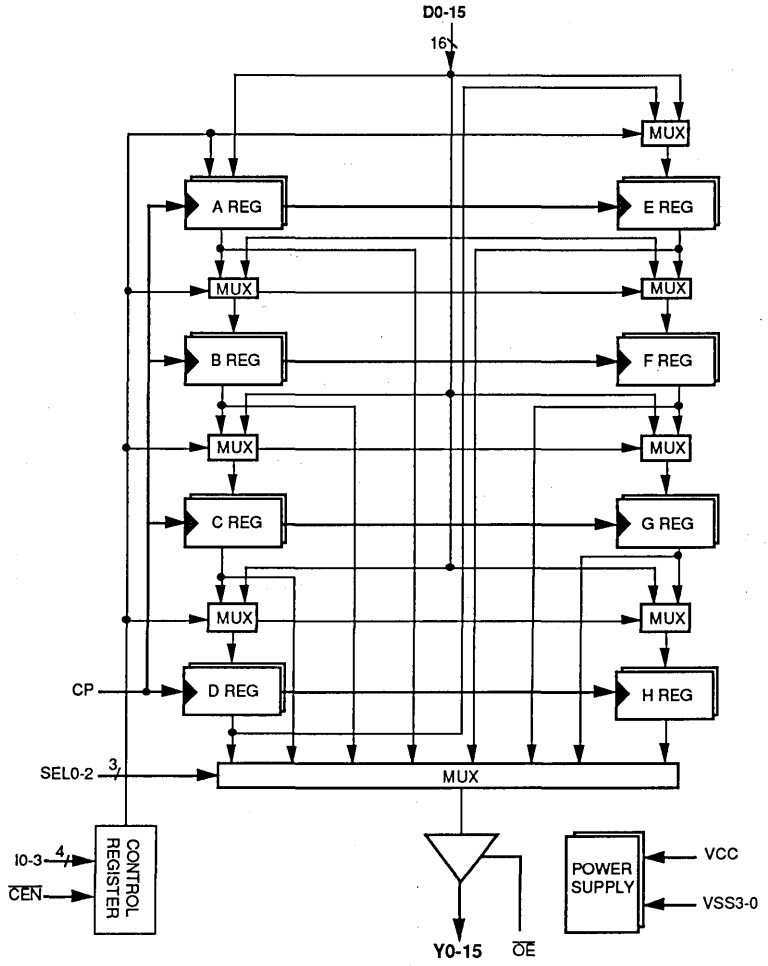
APPLICATION NOTE AN-65A

By Danh Le Ngoc, Ignacio Osorio and Avigdor Willenz

INTRODUCTION

The objective of this application note is to describe the use of the IDT 73200 multilevel pipeline register as the write buffer and read buffer for the R3000/1 RISC processor. The following topics are discussed:

- **The IDT73200 Multilevel Pipeline Register**, presents a brief description of general characteristics and configura tions of the multilevel pipeline register.
- **Read-Write buffers**, explains what read and write buffers are, and how they function in a R3000/1 system.
- **Implementing R-W Buffers**, describes how to implement the IDT73200 as read and write buffers. Buffer depths are also discussed in this section.
- **A Typical System**, provides an example of read-write buffers using the IDT73200, within a RISC system. It also presents the control logic and PAL equations to operate the IDT73200 as read and write buffers.



2647 drw 16

Figure 1. Block Diagram of the IDT73200

9

I3	I2	I1	I0	MNEMONIC	FUNCTION	PIPELINE LEVEL
0	0	0	0	LDA	D0-15->A	1
0	0	0	1	LDB	D0-15->B	1
0	0	1	0	LDC	D0-15->C	1
0	0	1	1	LDD	D0-15->D	1
0	1	0	0	LDE	D0-15->E	1
0	1	0	1	LDF	D0-15->F	1
0	1	1	0	LDG	D0-15->G	1
0	1	1	1	LDH	D0-15->H	1
1	0	0	0	LSHAH	D0-15->A->B->C->D->E->F->G->H	8
1	0	0	1	LSHAD	D0-15->A->B->C->D	4
1	0	1	0	LSHEH	D0-15->E->F->G->H	4
1	0	1	1	LSHAB	D0-15->A->B	2
1	1	0	0	LSHCD	D0-15->C->D	2
1	1	0	1	LSHEF	D0-15->E->F	2
1	1	1	0	LSHGH	D0-15->G->H	2
1	1	1	1	HOLD	HOLD ALL REGISTERS	

2647 drw 17 02

Figure 2. Load Control

THE IDT 73200 MULTILEVEL PIPELINE REGISTER

The IDT 73200 is a high-speed, low-power Programmable Multilevel Pipeline Register. It has a dedicated 16-bit input port and a dedicated 16-bit output port.

As shown in figure 1, the IDT73200 contains eight 16-bit registers which can be configured as one 8-level, two 4-level, four 2-level, or eight 1-level pipeline registers. Data at the input port D0-15 can be written into any of the eight registers under control of the load control: I0-3. Figure 2 illustrates the load control for the input port .

An eight-to-one output multiplexer allows data to be read on the Y-bus from any of the eight registers using the output-selection control: S0-2. Figure 3 illustrates the output control.

READ-WRITE BUFFERS

As shown in the Figure 4 , a high-speed computer system consists of a R3000/1 chip set, high-speed cache, write buffer, read buffer , I/O devices, and main memory. Since the main processor supports a write-through cache policy, all data written into the data cache must also be written into the main memory to maintain the cache coherency. Due to the data-rate mismatch between the high-speed processor bus (33MHz -> 240Mbytes/sec) and slow speed main memory (10-15MHz ->10-40 Mbytes/sec), a write buffer and a read-buffer are required. The write buffer is an elastic buffer which is used to capture addresses and data at the cache speed. At the other side of the write buffer, data is transfered into the main memory at the system memory speed.

When a load operation causes a cache miss, a main memory read is initiated. Two types of main memory read are supported on the R3000/1: single word transfer and multiple word transfer. In either case, a read-buffer is used to capture data from the system memory at memory speed. Then data is written into the cache at the cache speed. The depth of the write buffer and the read-buffer are dependent on different factors such as processor speed, system memory speed, bus protocol and the application.

SEL2	SEL1	SEL0	Y OUTPUT
0	0	0	A REG
0	0	1	B REG
0	1	0	C REG
0	1	1	D REG
1	0	0	E REG
1	0	1	F REG
1	1	0	G REG
1	1	1	H REG

2647 drw 18

Figure 3. Output Selection

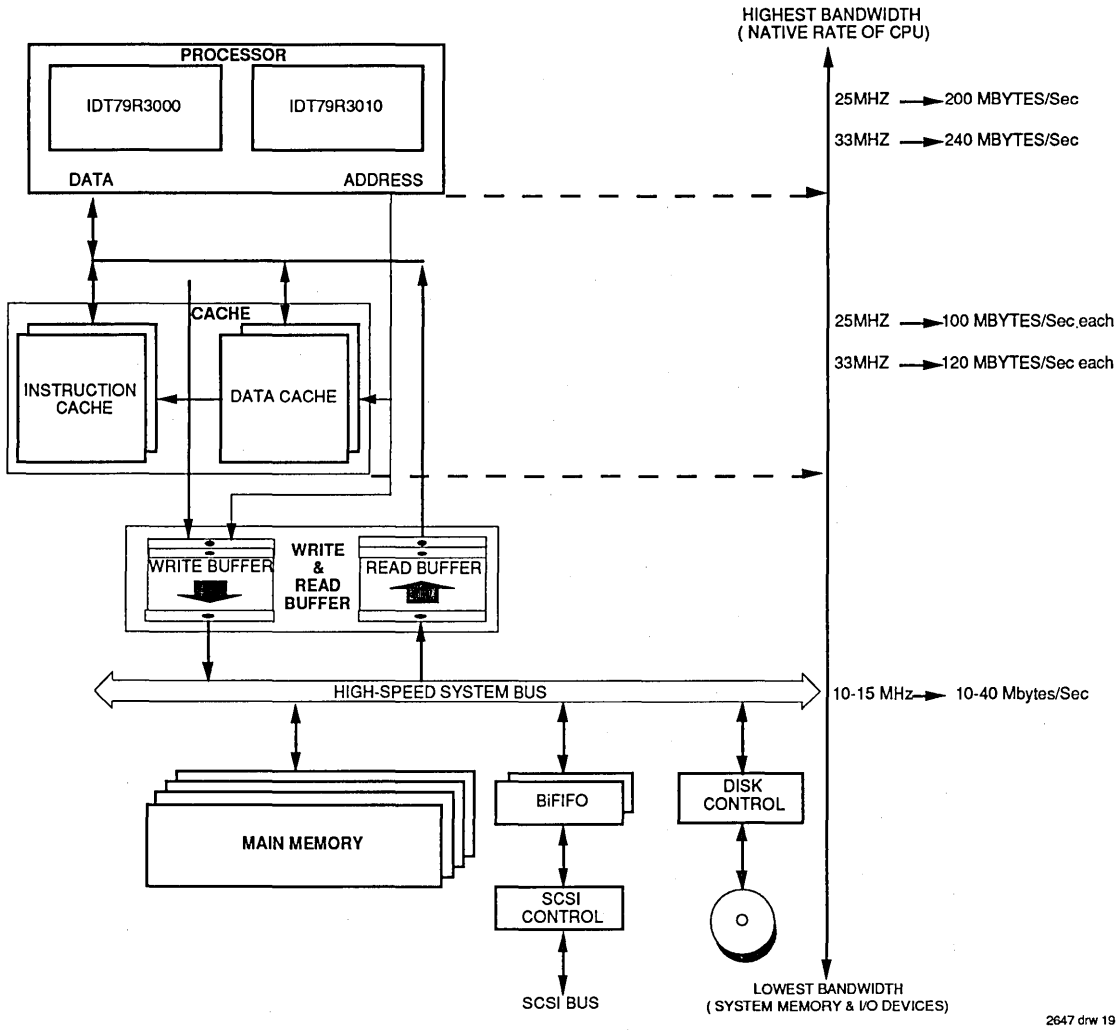


Figure 4. Simplified Block Diagram of a High-Speed RISC System

IMPLEMENTING R-W BUFFERS

As previously described in section 1, the IDT 73200 is like a high-speed synchronous memory with a depth that is programmable from 1 to 8 deep. Therefore, a write buffer and read buffer for the high performance R3000/1 system can be easily implemented with the help of the IDT 73200.

Figure 5 illustrates a detailed R3000/1 system which consists of the R3000/1 chip set, write buffer, read buffer, IDT49C465 Flow-thruEDC™, system main memory, and several state machines to control the main memory interface. In this scheme, the data bits together with the parity bits flow from the main memory through the EDC device for error detection and correction. When an error is detected, the EDC informs the read buffer control through an error feedback path.

In figure 5 the write buffer consists of two paths: address and data. The address path (34-bit) is created with three IDT 73200s to capture the address, tag, and the access type bits. The data path of the write buffer (32-bit), is formed by two IDT73200. Data coming from the CPU is buffered into the "data path" write buffer prior to being written into the main memory.

The read buffer in figure 5 consists only of a "data path" (36-bits) which includes the required data parity on the R3000/1 system. The IDT 49C465 high speed Flow-thruEDC can be used to maintain the data integrity of the system main memory. Also, parity bits are generated with the help of the IDT49C465 Flow-thruEDC.

9

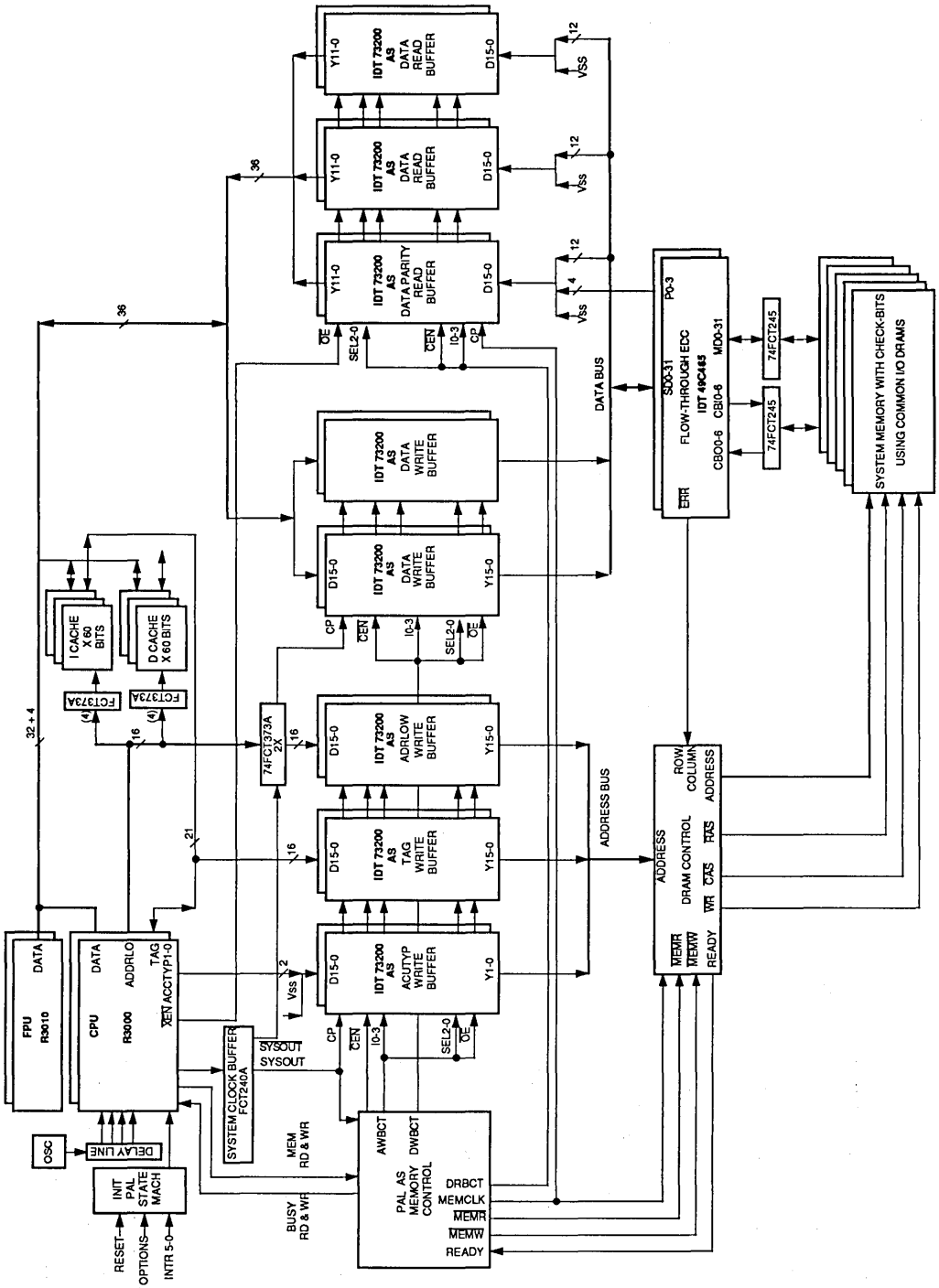


Figure 5. Detailed R3000 System with Read and Write Buffer

2647 drw 20

BUFFER DEPTH

As discussed earlier, the IDT 73200 can be configured for different depths: eight 1-level, four 2-level, two 4-level or one 8-level deep registers. This feature makes the 73200 particularly flexible in Read/Write buffer applications.

The depth of the write buffer is programmable using the load-control and the output selection. A single IDT 73200 can be programmed to a buffer depth of 1 to 8. A deeper write buffer can be implemented by cascading several devices in depth as shown in the figure 6. The right depth depends on the application program and/or hardware requirements.

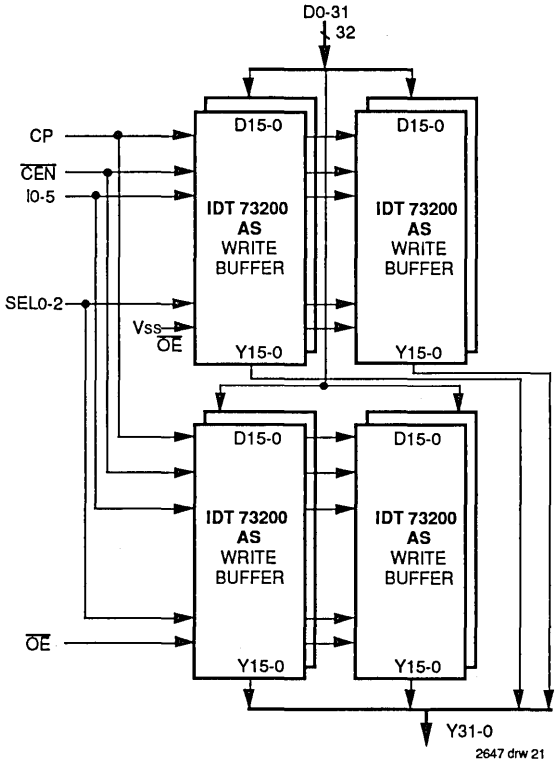


Figure 6. 16-Deep and 32-Wide Write Buffer Using IDT73200s

With the help of the write buffer, the CPU can write to the memory without regard to the memory speed. However, if consecutive (or back to back) write operations take place, the write buffer would eventually become full and cause the CPU to stall. Naturally, this could present a problem in high performance systems. A logical solution to CPU stalls is to increase the depth of the write buffer.

Typical write buffer depths are two and four levels. However, high end applications with intensive memory access may require deeper write buffers, i.e., eight or sixteen levels as shown in figure 6.

Read buffer depth design issues are somewhat different from those of the write buffers. For example, when an I-cache miss occurs we are faced with the question of "how many blocks to refill in the I-cache? " To answer this question, we should recall that the miss rate is determined by the cache size and the block size. Therefore, the block size determines the size of the read buffer. Thus to bring 16 words from memory into the I-cache would require a 16 level deep read buffer.

Fetching small blocks of instructions using a shallow read buffer implies constantly fetching instructions and therefore stalling the CPU for several cycles. Depending on the application, this could impose significant penalty on system performance. Due to program locality (sequentiality of instructions), we would benefit most by fetching a large block. Deep read buffers for I-cache are therefore an appealing solution. Typical read buffer depths are 4 levels; high end applications could consider from 8 to 16 levels read buffers.

D-cache fetching, on the other hand, is random in nature and typical schemes prefer a 1-level deep read buffer.

The flexibility of the 73200 allows instantaneous re-configuration when fetching for the I-cache and then for the D-cache. For example, we could have an R3000/1 initialized for 16 words I-cache fetching and 1 word D-cache fetching and still use the same read buffer. This can be accomplished through a read buffer controller capable of configuring the 73200 to different depths.

Lets now discuss two popular read buffer configurations.

a) 1-level deep Read Buffer &
b) 8-level deep Read Buffer

1-LEVEL DEEP READ BUFFER

One-level deep read buffers can be used in high performance systems where the data transfer between the main memory and the CPU is efficiently handled. This can be accomplished through a sophisticated memory scheme, like interleaving, and supported by a fast DRAM architecture. Such a scheme minimizes the transfer rate mismatch between the CPU and the main memory. One-level deep read buffers can also be applied in low performance systems where the penalty in fetching one word at a time is not significant.

8-LEVEL DEEP READ BUFFER

This configuration can be used in a general purpose system. An 8-level deep write buffer offers the benefit of effective data rate capture from the R3000/1 to the main memory. The 8-level Read Buffer is convenient for slow main memory systems.

9

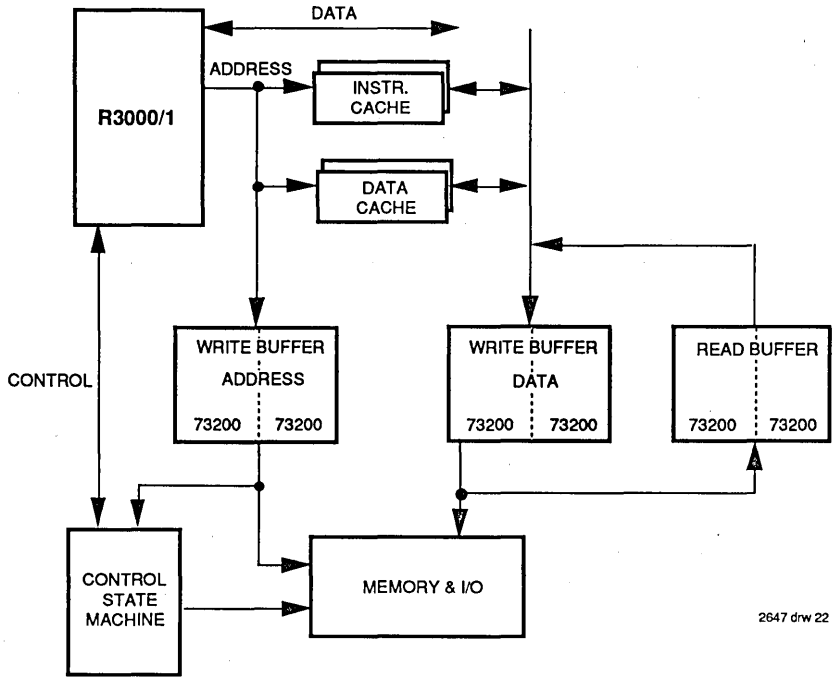


Figure 7. R3000/1 System with the Read-Write Buffers

A TYPICAL SYSTEM

Figure 7 shows the interconnections among the R3000/1, Instruction and Data Cache, Read/Write Buffers, control state machine, and the system memory. The read and write buffers are built from multilevel pipeline registers denoted by the IDT 73200. The control state machine represents the logic needed to drive the read and write buffers.

WRITE BUFFER INTERFACE

A write buffer, as discussed earlier, transfers data from cache to main memory and provides address bits to select memory locations. This is illustrated in Figure 8: one write buffer is dedicated to pass address bits and the other transfers data to the main memory. Therefore, the write buffer labeled "address" is activated in both memory reading and memory writing operations.

As seen in Figure 8, the address path carries address, tag and Acc type bits. Notice that the write buffer labeled "Address", is formed by two IDT 73200. The first 73200 captures Address low 0-13 and AccTyp 0,1. The second 73200 captures Adr High 14-29 and Tag 0,1.

The Data path, as shown in figure 8, carries data and parity bits. The data write buffer uses two IDT 73200. They latch 32 data bits from the cache and transfer them to a memory location selected by a memory controller. Notice that parity bits can be generated using the IDT49C465 when data is flowing from the write buffer to the system memory.

A situation of interest in deep write buffers is the following: The CPU requests reading data from a memory location that is about to be updated by the write buffer. The potential problem is clear: reading data that hasn't been updated yet. To avoid this problem, write buffer systems use conflict checking schemes.

A common "conflict checking scheme" is implemented by comparing addresses of memory locations to be read and written by the read/write buffers. When an address match is found, a match signal is send to the CPU. This solution may involve using more hardware to implement such scheme. Another approach is "flushing". To simplify the design, the write buffer is "flushed", i.e., all pending writings are placed in the main memory before any read buffer operation takes place. Such is the case in figure 8 where no additional hardware was needed.

Figure 8 shows the associated control circuitry to drive the write buffer. Notice that write buffer "data" and AdrHi (14-31) are clocked at the $\overline{\text{SYSOUT}}$ signal, whereas AccTyp (0:1) and adr-Lo(0:13) are clocked at $\overline{\text{SYSOUT}}$.

WRITE BUFFER CONTROLLER

The write buffer controller is internally driven by two counters: The I-counter selects the load operation for the input to the 73200 . The SEL-counter selects the register to be read in the 73200 output. The write buffer controller also takes care of the "flushing" scheme.

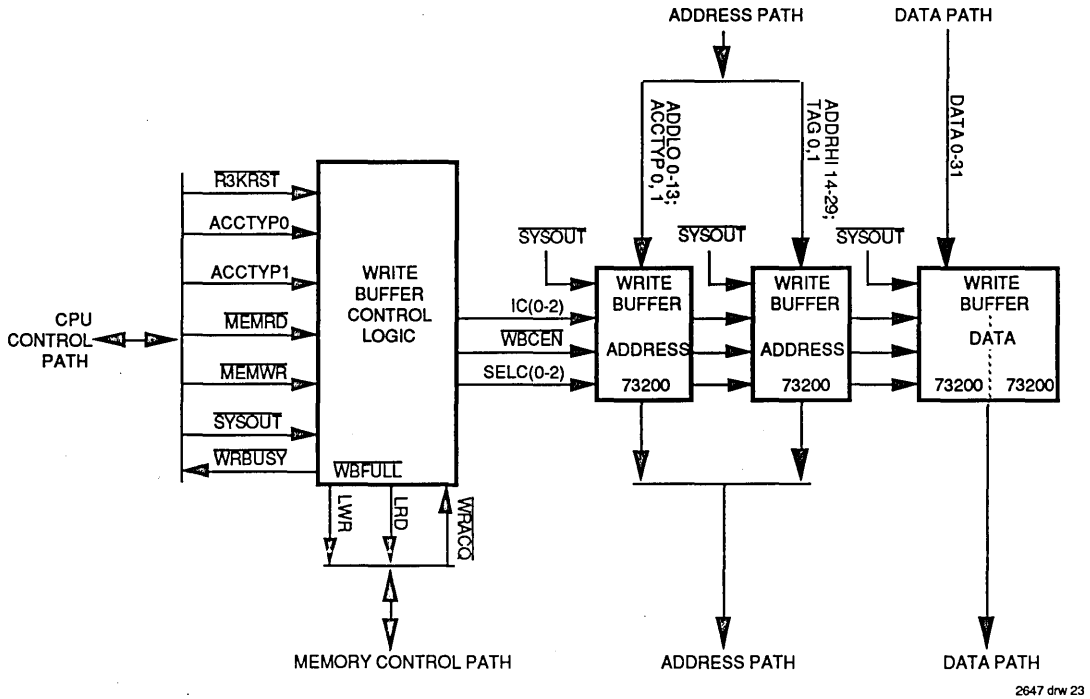


Figure 8. Write Buffer Interface

The PAL equations for the write buffer controller are:

```
MODULE WB_CONT;

TITLE WB_CONT;
TYPE MMI 16R4;
Inputs;
        IC3             Node[pin2];
        SELC3           Node[pin3];
        WBEMPTY         Node[pin4];
        WRACQ           Node[pin4];
        MEMRD           Node[pin5];
        WBFULL          Node[pin6];
        MEMWR           Node[pin7];
        RESET           Node[pin9];

        LRD             Node[pin15];
        LWR             Node[pin14];

Outputs;
        LRD             Node[pin15];
        LWR             Node[pin14];
        X               Node[pin19];
        WBCEN       Node[pin12];
        ICE         Node[pin13];
        SELCE       Node[pin18];

Table;
        X NOT =         ICE3 XOR SELC3;
        LWR NOT:=       LRD AND WBEMPTY AND WRACQ
                        AND RESET OR LRD AND!MEMWR
                        AND WRACQ AND! WBEMPTY AND
                        RESET;

        LRD NOT :=      (!WBEMPTY AND !MEMRD AND
                        LWR AND RESET) OR (!LRD AND
                        !MEMRD AND RESET);

        WBCEN NOT =     WBFULL;

        ICE NOT =       (!MEMWR AND WBFULL)
                        OR(!MEMRD AND !WBEMPTY);

        SELCE NOT=      (!LWR AND !WRACQ) OR
                        (!LRD AND MEMRD);
END;
END WB_CONT.
```

9

The PAL equations for the I-counter are:

```
MODULE I-COUNTER;
TITLE I_COUNTER;
TYPE MMI 16R4;

Inputs;
        X                   Node[pin2];
        IC                  Node[pin3];
        /ICE                Node[pin4];
        /R3KRST             Node[pin5];
        IC3                 Node[pin17];
        IC2                 Node[pin16];
        IC1                 Node[pin15];
        IC0                 Node[pin14];
Outputs;
        IC3                 Node[pin17];
        IC2                 Node[pin16];
        IC1                 Node[pin15];
        IC0                 Node[pin14];
        /WBFULL             Node[pin19];
        /WBEMPTY            Node[pin12];
Table;
        IC0 NOT :=          ( IC0 AND !/ICE ) OR (!IC0
                            AND /ICE) OR !/R3KRST;

        IC1 NOT:=           (!IC0 AND !IC1 AND !IC1AND
                            !/ICE) OR (IC0 AND IC1 AND
                            !/ICE) OR (!IC1 AND /ICE) OR
                            !/R3KRST;

        IC2 NOT:=           (!IC0 AND !IC1 ABD !IC2
                            ABD !/ICE) OR (IC0 AND IC1
                            AND IC2 AND !/ICE) OR
                            (!IC0 AND IC1 AND !IC2 AND
                            !/ICE) OR (IC0 AND !IC1
                            AND !IC2 AND !/ICE) OR
                            (!IC2 AND /ICE) OR
                            !/R3KRST;

        IC3 NOT :=          (!IC2 AND !IC3 AND !/ICE)
                            OR (!IC1 AND IC2 AND !IC3
                            AND !/ICE) OR (!IC0 AND
                            IC1 AND IC2 AND !IC3 AND
                            !/ICE) OR (IC0 AND IC1 AND
                            IC2 AND IC3 AND !/ICE) OR
                            (!IC3 AND /ICE) OR
                            !/R3KRST;
        /WBFULL NOT =  !IC AND !X;
        /WEMPTY NOT=   !IC AND X;
End;
End I_Counter;
```

The PAL equations for the SEL-counter are:

```
MODULE SEL_COUNTER;
TITLE SEL_COUNTER;
TYPEMMI 16R4;
Inputs;
        /SELCE              Node[pin2];
        IC0                 Node[pin3];
        IC1                 Node[pin4];
        IC2                 Node[pin5];
        /R3KRST             Node[pin6];
        SELC3               Node[pin17];
        SELC2               Node[pin16];
        SELC1               Node[pin15];
        SELC0               Node[pin14];

        S0                  Node[pin18];
        S1                  Node[pin13];
        S2                  Node[pin19];
Outputs;
        S0                  Node[pin18];
        S1                  Node[pin13];
        SELC3               Node[pin17];
        SELC2               Node[pin16];
        SELC1               Node[pin15];
        C0                  Node[pin14];
        S2                  Node[pin19];
        C                   Node[pin12];
Table;
        SELC0 NOT:=         (SELC0 AND !/SELCE) OR
                            (!SELC0 AND /SELCE) OR
                            !/R3KRST;

        SELC1 NOT:=         (!SELC0 AND !SELC1 AND
                            !/SELCE) OR (SELC0 AND
                            SELC1 AND /SELCE) OR
                            !/R3KRST;

        SELC2 NOT:=         (!SELC0 AND !SELC1 AND
                            !SELC2 AND !/SELCE) OR
                            (SELC0 AND SELC1 AND
                            SELC2 AND !/SELCE) OR
                            (!SELC0 AND SELC1 AND
                            !SELC2 AND !/SELCE) OR
                            (SELC0 AND !SELC1 AND
                            !SELC2 AND !/SELCE) OR
                            (!SELC2 AND /SELCE) OR
                            !/R3KRST;

        SELC3 NOT:=         (!SELC2 AND !SELC3 AND
                            !/SELCE) OR (!SELC1 AND
                            SELC2 AND !SELC3 AND
                            !/SELCE) OR (!SELC0 AND
                            SELC1 AND SELC2 AND
                            !SELC3 AND !/SELCE) OR
                            (SELC0 AND SELC1 AND
                            SELC2 AND SELC3 AND
                            !/SELCE) OR (!SELC3 AND
                            /SELCE) OR !/R3KRST;

        S0 NOT =            (IC0 XOR SELCO);
        S1 NOT =            (IC1 XOR SELC1);
        S2 NOT =            (IC2 XOR SELC2);

C NOT = S0 AND S1 AND S2;
End;
End SEL_Counter;
```

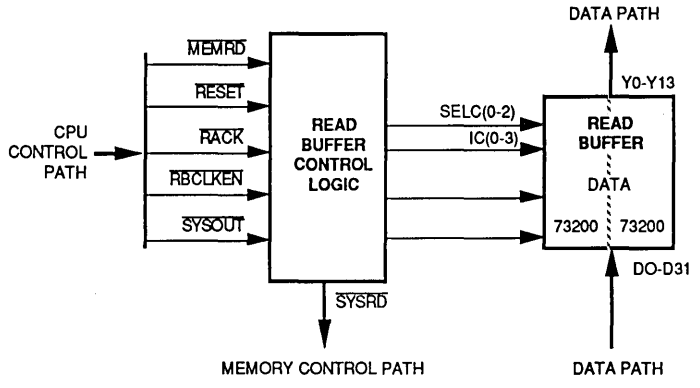


Figure 9. Read Buffer Interface

READ BUFFER INTERFACE

When reading from the main memory to the cache, the R3000/1 sends a memory read signal to the control state machine, represented in Figure 9 as the Read Buffer Control Logic. Once the signal has been acknowledged, the R3000/1 places the address, tag, and data size in the write buffers. Internally, the 73200 registers capture this information at R3000/1 clock rate with load and output configurations determined by the read buffer controller. Once the address is available in the address bus, the controller will then drive memory signals to initiate the memory transfer at memory clock rate into the read buffer.

READ BUFFER CONTROLLER

The read buffer controller monitors the flow of data within the Read Buffer by programming the 73200 internal registers to the appropriate load mode and memory clock frequency. Finally, the controller selects the output registers at such speed to match the R3000/1 frequency.

The PAL equations for the read buffer controller are:

```
MODULE W/RB_CONT;
TITLE W/RB_CONT;
TYPE MMI 16R8;
Inputs;
        WRACQ           Node[pin5];
        MEMRD           Node[pin6];
        MEMWR           Node[pin8];
        RESET           Node[pin9];
        LRD             Node[pin15];
        LWR             Node[pin14];
        WB_CLK_DIS      Node[pin19];
        CMEMRD          Node[pin13];
        CCMEMRD         Node[pin18];
        WRBUSY          Node[pin16];
Outputs;
        LRD             Node[pin15];
        LWR             Node[pin14];
        WB_CLK_DIS      Node[pin19];
        WB_DATA_OE      Node[pin12];
        CMEMRD          Node[pin13];
        CCMEMRD         Node[pin18];
        WRBUSY          Node[pin16];
Table;
        LWR NOT :=      LRD AND !MEMWR AND
                        RESET OR !LWR AND
                        WRACQ;
        LRD NOT :=      LWR AND !MEMRD;

        WRBUSY NOT :=   !MEMWR OR
                        !WRBUSY AND WRACQ
                        OR !MEMRD;

        WBCEN NOT :=    !MEMWR OR !WB_CLK_DIS
                        AND WRACQ OR !MEMRD
                        AND CCMEMRD;

        WB_DATA_OE NOT:= LRD AND !MEMWR OR
                        {need to be inverted}
                        !LWR AND WRACQ;

        CMEMRD NOT :=   !MEMRD;

        CCMEMRD NOT:=   CMEMRD;
End;
End W/RB_CONT;
```

CONCLUSION

As the speed of the processor increases, write- and read buffers must also become faster and deeper. The high-speed multi-level pipeline register IDT 73200 meets that challenge by providing a fast and flexible data path to suit various high-speed RISC and CISC processors.

9

Integrated Device Technology, Inc.

USING IDT73210 AS READ AND WRITE BUFFERS WITH R3000

APPLICATION NOTE AN–65B

By V. S. Ramaprasad

INTRODUCTION

In this application note, the design of one deep read and one deep write buffer to be used in an R3000 system is described with boolean equations and timing diagrams. The boolean equations are for the control signals of the read and write buffers and the main memory interface. This control logic can be implemented with any PLD. The syntax chosen to describe these equations is simple and it is not associated with any PLD programming software. The timing diagrams explain the various states during the operation of one deep read and write buffers. Also described in this application note are the other possible configurations of implementing read and write buffers with IDT73210s. These components can be used as two deep read and one deep write , and one deep read and two deep write buffers. Before the application is presented, the features of 73210 are described and a summary of the memory interface signals of R3000 is given.

R3000 based systems require read/write buffers between the CPU and the main memory due to memory bandwidth mismatch. The main memory system supplies the instructions/ data through a read buffer. The CPU makes the data updates to the main memory through a write buffer. The speed differences between the CPU, the caches and the main memory that typically exist in many systems demand the use of at least one level deep read and write buffers. The use of these buffers isolates the caches from the rest of the memory system. They also limit the physical length of the address and data lines and serve as drivers to the rest of the system.

The gain in performance by increasing the depth of the read and the write buffers is completely dependent on the application program being executed. By modeling memory subsystems with different depths of read/write buffers (using the System Programmer's Package tools for the R3000) and running the application program on the model, the designer can make the trade-off between the cost and the depth of the buffers. For high performance systems with sophisticated main memory schemes like interleaving, and for systems with fast DRAM architectures like Page Mode, or Static Column Mode, a one deep read buffer might satisfy the transfer rate of the processor. For low performance systems, where the penalty of fetching one word at a time is not significant, and for applications with infrequent successive writes, a one deep write buffer might also deliver optimal performance.

In systems where one-level deep read and write buffers proved to be sufficient, a bidirectional register can be utilized to serve as both read and write buffers. The 8-bit bidirectional register, IDT73210, with parity checking and parity generation is an ideal candidate for this purpose. This bidirectional register also allows the designer to build a two-level deep read buffer and one level deep write buffer, or one- level read buffer and two-level write buffer configurations. Using IDT73210 reduces the parts that are needed for parity generation. Also, by clocking in the lower address bits and the higher address bits with separate clocks, the designer can eliminate latching the address low bits.

IDT73210 FEATURES

Figure 1 shows the features of IDT73210 with all the control signals and data paths. It is a bidirectional buffer with separate output enables and clock enables. Data is registered with the same clock in both directions. There is a single data path from port A to port B. The 8-bit data and the parity bit are clocked through register X. The POLARITY signal is used to select even or odd parity generation. Even parity checking is done on the data, and a parity error is indicated by PERRA. The 8-bit data and the parity bit are enabled through a tri-stateable buffer to port B.

There are two data paths from port B to port A. A multiplexer controlled by SEL selects a path. Even parity checking is done in both the paths and parity error is indicated by PERRB. The first path is through latch W and register Z. In this path bit W8 is complemented by POLARITY to yield either even or odd parity.The second path is through registers Y & Z and even parity is generated on the data. The two registers in the second data path provide the user with two- level deep buffering. The 9-bit output is enabled through a tri-stateable buffer to port A.

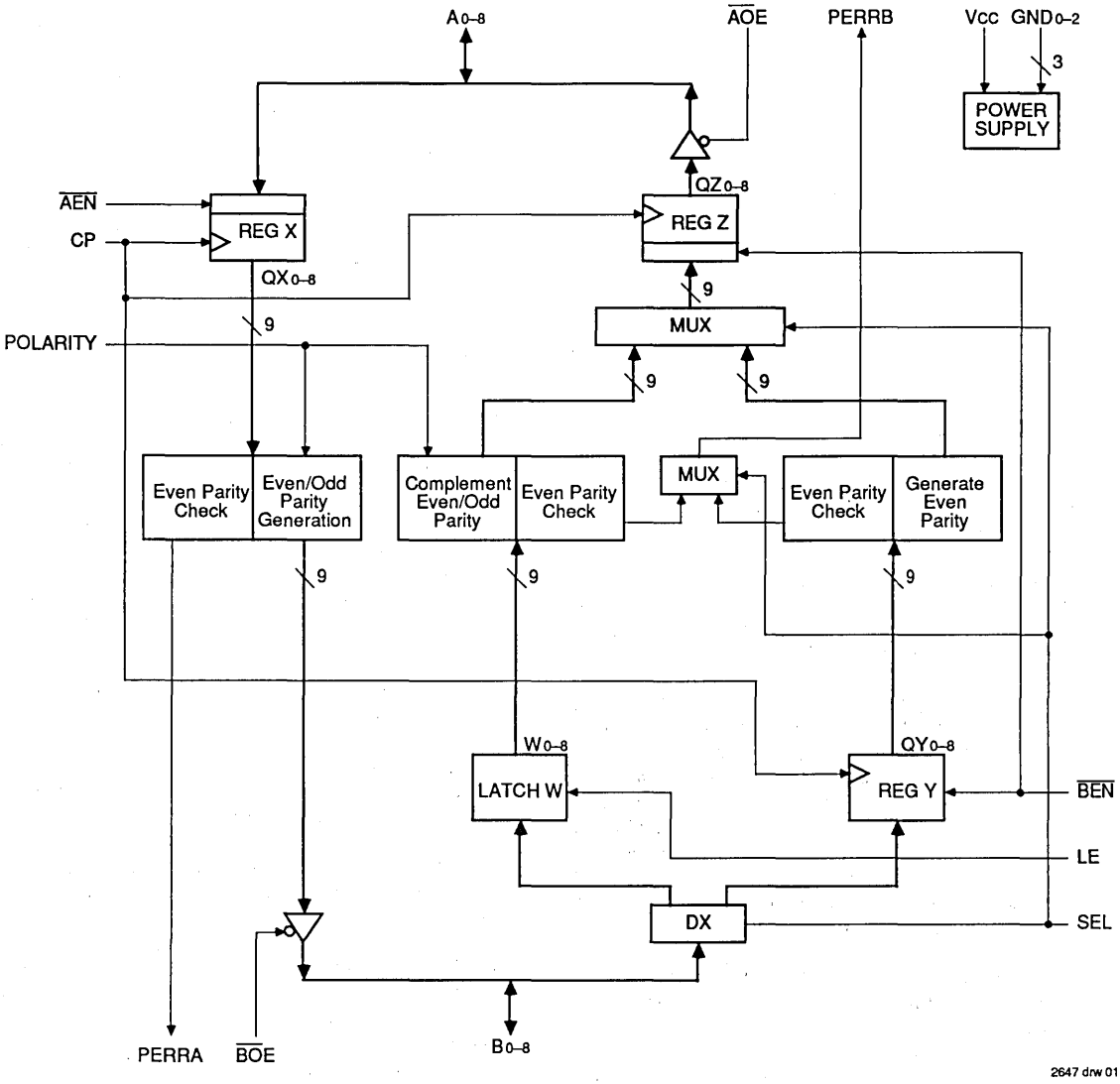


Figure 1. IDT73210

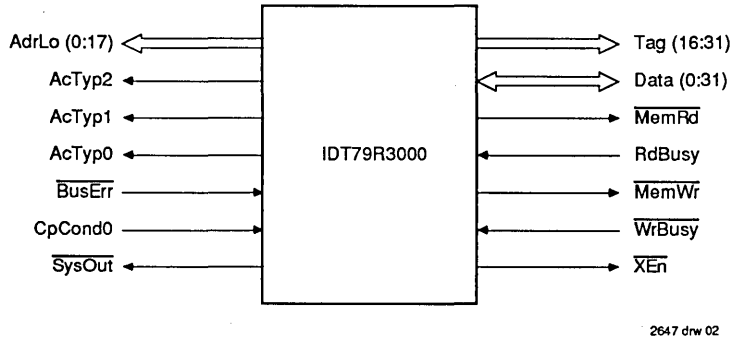


Figure 2. Memory Interface Signals

IDT79R3000 MEMORY INTERFACE

The R3000 has interfaces to the main memory through the asynchronous memory bus. The output signals indicate the nature of operation that the R3000 is performing. The input signals are used to indicate the termination of a stall, block refills, and to cause exception processing.

The figure above shows the signals used to interface to main memory. The address bus is split into AdrHi and AdrLo. The AdrHi bus is also used as the Tag bus for cache reads and therefore is shown as bidirectional.

MemRd: This signal indicates the entry into the stall on a read operation. It is an active-low output signal. This output signal of the R3000 is used by the state machines to enter a read state and signal the memory system that the R3000 accepts data from the supplied 32-bit address. For one word refill, MemRd is deasserted by the R3000 one cycle after the RdBusy signal is deasserted indicating that the required data is ready. The deassertion of MemRd signals the end of a read stall. MemRd stays asserted during the entire stall cycles.

RdBusy: This input signal to the R3000 is used to enter and terminate read stall cycles. The deassertion of RdBusy terminates stall cycles and the R3000 enters a fixup one cycle later during single word loads or it enters refill cycles in case of multiple word loads. RdBusy assertion and deassertion is sampled by the R3000 in phase 1 of the clock cycle.

XEn: This active-low, output signal is used to enable the output of the read buffer in refill and fixup cycles.

MemWr: This output signal is asserted low for store operations. Unlike MemRd, this signal is active for only one cycle as are the associated data and addresses. MemWr is used to enter a write state.

WrBusy: In order to create a write stall, this input signal to the R3000 has to be asserted low during the cycle in which MemWr is asserted. The deassertion of WrBusy terminates a write stall and the R3000 enters the fixup cycle. In the fixup cycle, the last write operation during which WrBusy was asserted is repeated. WrBusy is usually tied to the signal that indicates the write buffer is full. WrBusy assertion is sampled by the processor in phase 2 and the deassertion is sampled in phase 1 of the clock cycle.

SysOut: This is the clock output of the 79R3000 and is the clock frequency at which the R3000 is rated.

CpCond0: The condition of this input signal to the R3000 in stall cycles determines if the processor will do a single word read or a multiple word read.

BusErr: This input signal is provided as a mechanism to create an exception in the R3000 and as an aid to escape from interminable stall cycles.

AccTyp0: This output signal has three functions. During cached reads it indicates whether there was a data cache miss or an instruction cache miss. This information is useful if the block refill size is different for data and instructions. During uncached reads it is used with AccTyp1 to indicate the size of the data being read. During writes it is used along with the AccTyp1 to indicate the size of the data being written.

AccTyp1: This output signal is undefined for cached reads. For uncached read operations and for store operations, AccTyp1 along with AccTyp0, indicates the size of data transfer.

AccTyp2: AccTyp2 is undefined for store operations with stall cycles. For load operations, it is high for cached operations and low for uncached operations. During run cycles, this line indicates whether there is any data transfer during the second phase.

USING IDT73210s

Figure 3 shows the application of the 73210s as one-deep read and one-deep write buffer. Four 73210s are used to transfer the 32-bit data and the associated four parity bits. On the address bus, four 73210s are used to pass the 32-bit physical address and the access type bits(0:1) to the main memory. Port B of the 73210s are connected to the processor side, and Port A of the 73210s are connected to the memory side.

The read and the write data paths are explained in Figures 4 and 5. In this design, one single set of four IDT73210s serve the function of read and write buffers. Also, a set of four IDT73210s are used to capture the addresses during read and write operations. The timing diagrams point out the control signals that resolve any conflicts in the use of these buffers.

The control logic, described in the following sections, can be implemented with any PLD that matches the processor speed. To interface with the main memory, signals are defined to make a request to the main memory ($\overline{MREQ}$), to specify a read or a write operation to the main memory ($\overline{MRD}$, $\overline{MWR}$), and a signal from the main memory to indicate the completion of read or write operation ($\overline{CYCEND}$).

The memory interface signals from the R3000 are used by the PAL state machine inorder to generate controls to the buffers and the main memory. The RdBusy, $\overline{WrBusy}$, $\overline{MREQ}$, $\overline{MRD}$, $\overline{MWR}$, and the clock and data output enables for the 73210s are generated by the state machine.

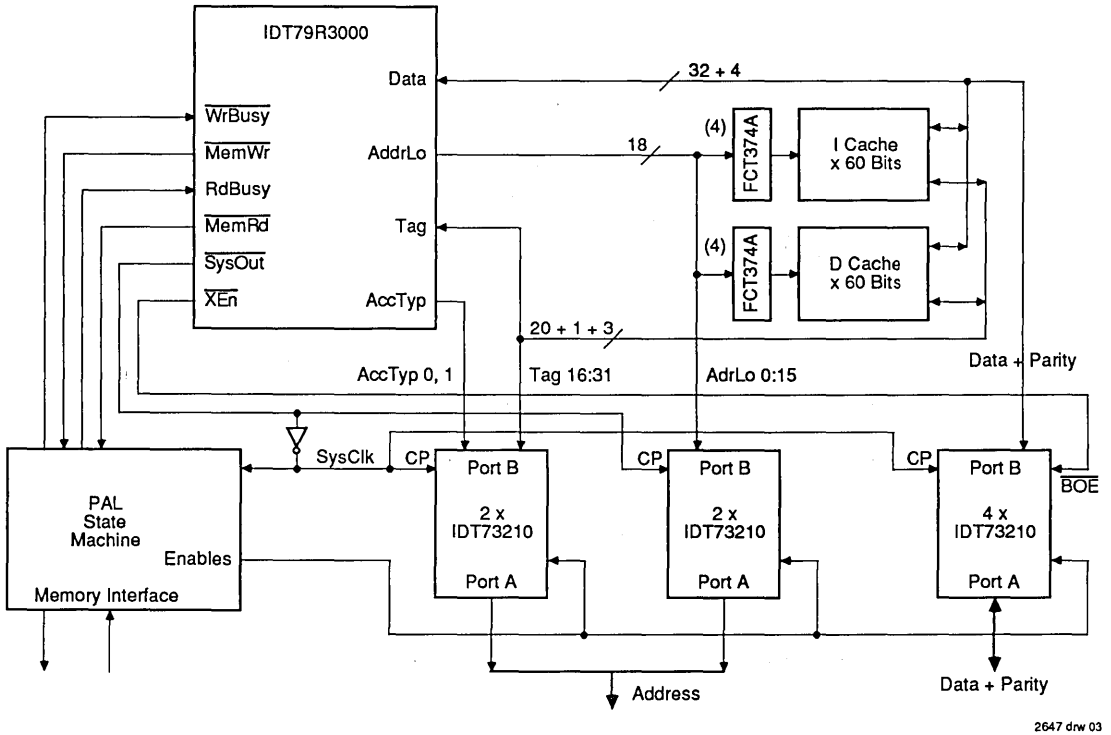


Figure 3. Using IDT73210 as Read and Write Buffer

9

Read Operations

The data path for the read operations is through register X. The address and the access type bits go through the latch W and the register Z. The lower address bits are clocked in with $\overline{SysOut}$. The tag bits, along with the access type bits, are registered with inverted $\overline{SysOut}$ (SysClk). The latch W is always transparent to bypass the address bits and access type bits. The POLARITY signal is held low to pass the access type bits as parity bits through the two 73210 on the tag bus. The low POLARITY signal to the four 73210s on the data bus generates even parity on the data passing through register X.

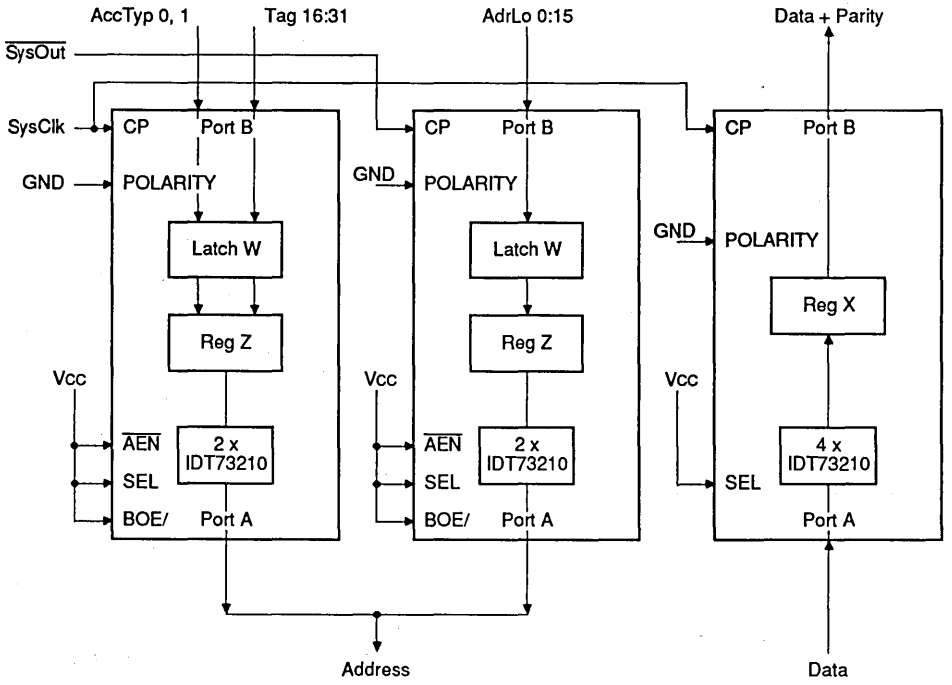


Figure 4. Read Data Path. One Deep Read, One Deep Write Buffers Using IDT73210

Write Operations

The data path for write operations is through latch W and register Z. Data is clocked in the 73210s on the data bus with SysClk along with the tag bits. The lower address bits are clocked in with $\overline{\text{SysOut}}$. Once the address and data are available in the Z registers, the PAL state machine generates the output enable signals and presents the address and the data to the memory. The even parity that is generated by the CPU passes through the parity unit without getting modified.

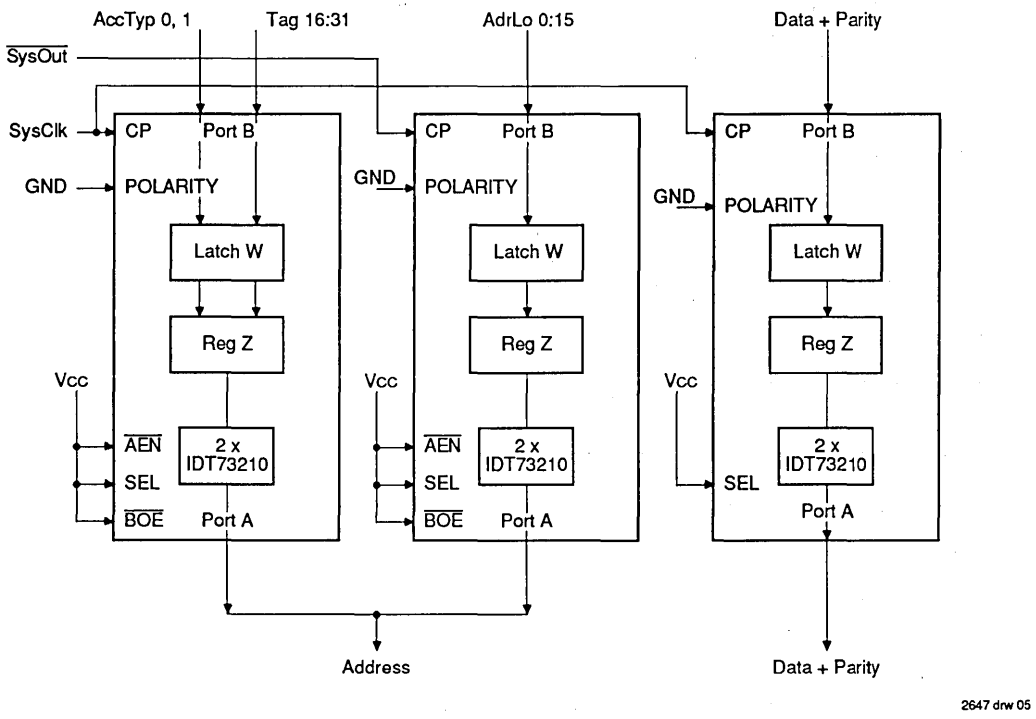


Figure 5. Write Data Path. One Deep Read, One Deep Write Buffers Using IDT73210

CONTROL LOGIC

The control logic for the signals that control the address 73210s, the data 73210s, and handshake signals to the main memory system is described with simple boolean equations. The 73210s are used to capture the data and addresses during read and write operations and to provide the system with one-level deep read and write data paths. Byte enable signals for partial word writes can be generated by extending these equations. In this design, block refills are supported and instruction streaming is assumed to be enabled.

The control signals that are utilized by the PAL state machine to control the read/write buffers, and communicate with the main memory controller are $\overline{\text{WIP}}$, $\overline{\text{WrBusy}}$, RdBusy, $\overline{\text{MRD}}$, $\overline{\text{MWR}}$, $\overline{\text{MREQ}}$ and $\overline{\text{CYCEND}}$. The clock input to the PAL is inverted $\overline{\text{SysOut}}$. In the following boolean equations the following notation is adopted:

!	Logical NOT operation.
*	Indicates the corresponding signal is active low.
OR	Logical OR operation.
AND	Logical AND operation.
;	To end a boolean equation.
:=	Registered output.
=	Combinatorial output.

9

Main Memory Controls

```
{ RdBusy usually stays asserted even when there is no read                }
{ operation going on. It is deasserted when the memory system             }
{ acknowledges ( asserting CycEnd) that the read is finished. It          }
{ remains deasserted till MemRd gets deasserted.                          }
{ It is assumed here that the memory system asserts CYCEND in             }
{ phase 1. RdBusy gets deasserted in phase 2, and the CPU puts            }
{ out XEn(s) from the next clock cycle.                                   }
{ RdBusy is not deasserted with CYCEND associated with a prior            }
{ write operation.                                                        }
{ RdBusy is a registered output.                                          }

RdBusy :=  !( !WIP AND MemRd AND CYCEND) OR
           !( !RdBusy AND MemRd);

{A single pulse request is sent out to the main memory system to          }
{ indicate that a read or write operation is coming along. It is          }
{ asserted only when a read or write operation is feasible through        }
{the one deep read and write buffers.                                     }
{ This is a registered output.                                            }

MREQ :=  ( !WIP AND !MemRd AND MemWr AND !MREQ) OR
         ( !WIP AND MemRd AND !MREQ );

{ A read strobe is given out to the main memory system to                 }
{ indicate a read operation. This signal is asserted while there is       }
{ no write in progress and MemRd is asserted.                             }
{ To support block refills MRD stays asserted with MemRd.                 }
{ This is a registered output.                                            }

MRD :=  ( !MRD AND !WIP AND MemRd) OR
        ( MRD AND MemRd);
```

The control signal CYCEND is asserted by the main memory controller to indicate the finish of a write operation or the availability of the first word of the block refill in the read buffer.

It is assumed that CYCEND is asserted in phase 1, so that RdBusy can also be deasserted in phase 1.

```
{ WIP signal is used to indicate whether the write buffer is in           }
{ the process of retiring its contents to the main memory.                }
{ WIPis asserted when MemWr is asserted and deasserted when               }
{ an acknowledge (CYCEND) from the main memory system comes               }
{ back indicating that the write is carried out.                          }
{ WIP is a registered output                                              }

WIP :=  ( !WIP AND !MemRd AND MemWr) OR
        ( WIP AND !CYCEND);

{ Write busy (WrBusy) is asserted when there is a write in                }
{ progress or when read operation is going on. During read                }
{ operations with streaming enabled, WrBusy should be                     }
```

```
{ asserted to stop any writes because there is a common data      }
{ buffer for both read and writes.                                }

WrBusy = WIP OR MemRd;

{ A write strobe is given out to the main memory system to        }
{ indicate a write operation is in progress.                      }

MWR = WIP;
```

Higher Address Buffer Controls

```
{ Controls for 73210s that pass higher address bits (Tag 16:31)   }
{ and access type bits (AccTyp 0,1).                              }
{ The path through latch W & register Z is selected by the internal }
{ Mux.                                                            }

SEL = 1;

{ Register Z is enabled for read and write operations when there  }
{ is no contention between read and writes.Latch W is transparent. }
{ A read operation in progress is indicated by MemRd signal.      }
{ For write operations ABEN is enabled for one clock cycle.       }

DMemWr := MemWr;
ABEN = ( MemRd AND !WIP AND !MemWr) OR
       ( MemWr AND !WIP AND !MemRd) OR
       ( ABEN AND MemWr AND !DMemWr);

{ Allows the Access Type bits to pass through "Compliment         }
{ Even/Odd Parity" unit as parity bits without getting modified.  }

POLARITY = 0;

{ The higher address bits along with the access type bits are     }
{ clocked into the register Z with inverted SysOut.               }

CP = ! SysOut;

{ The higher address bits along with the access type bits are put }
{ out to port A when there is a write in progress or while MemRd  }
{ is asserted.                                                    }

AAOE = WIP OR MemRd;

{ AEN always disabled for the address 73210.                      }

AEN = 1;

{ BOE always disabled for the address 73210.                      }

BOE = 1;
```

Lower Address Buffer Controls

```
{ Controls for 73210s that pass lower address bits (AddrLo 0:15)  }
```

```
{ The path through latch W & register Z is selected by the internal   }
{ Mux.                                                                }

SEL = 1;

{ Register Z is enabled for read and write operations when there      }
{ is no contention between read and writes.Latch W is transparent.    }
{ A read operation in progress is indicated by MemRd signal.          }
{ For write operations ABEN,is enabled for one clock cycle.           }

DMemWr := MemWr;
ABEN = ( MemRd AND !WIP AND !MemWr) OR
       ( MemWr AND !WIP AND !MemRd) OR
       ( ABEN AND MemWr AND !DMemWr);

{ POLARITY is Don't Care.                                             }

POLARITY = 0;

{ The lower address bits are clocked into the register Z with         }
{ SysOut signal, because they are available in the first phase.       }

CP = SysOut;

{ The lower address bits are put out to port A when there is a        }
{ write in progress or while MemRd is asserted.                       }

AAOE = WIP OR MemRd;

{ AEN always disabled for the address 73210.                          }

AEN = 1;

{ BOE always disabled for the address 73210.                          }

BOE = 1;
```

Data Buffer Controls

```
{ Controls for 73210s that transfer data bits for reads & writes.     }
{ The path through latch W & register Z is selected by the internal   }
{ Mux to provide one-deep write buffer.                               }

SEL = 1;

{ Register Z is enabled for write operations when there is no         }
{ read operation in progress. A read operation in progress is         }
{ indicated by MemRd signal. Latch W is transparent.                  }

DMemWr := MemWr;
DBEN = ( MemWr AND !WIP AND !MemRd) OR
       ( DBEN AND MemWr AND !DMemWr);

{ Even polarity generated by the CPU is passed through by setting     }
{ POLARITY to ZERO.                                                   }
```

POLARITY = 0;

{ The data bits along with the parity bits are clocked into the }
{ register Z with inverted SysOut. }

CP = ! SysOut;

{ The data bits are put out to port A when there is a write in }
{ progress. }

DAOE = WIP;

{ DAEN is enabled during read operations. }

DAEN = MemRd AND !WIP;

DBOE is enabled by XEn to read the data from the read buffer.

TIMING DIAGRAMS

Figures 6 through 11 give the timing waveforms for the one-deep read and one-deep write buffer described in Figure 3. The signals shown in these figures are described by the boolean equations presented earlier. In these timing diagrams, the signals that are generated by the PAL state machine are shown with a displacement in relation to their input signals. Also, some of the signals generated by the PAL state machine are registered with SysClk. The main memory interface signals generated by the PAL are MREQ, MRD, CYCEND, and MWR. The enable signals to the address 73210s are ABEN, and AAOE. The enable signals to the data 73210s are DBEN, DAOE, DAEN, and DBOE. The memory acknowledge signal, CYCEND is asserted two cycles after MREQ is asserted, for both read and write operations.

Figure 6 shows read and write operations. The memory read operation starts with the MemRd signal being asserted. A MREQ pulse is sent out to the memory, and MRD signal is asserted for the duration that the MemRd signal stays asserted. The memory system responds to the request by placing the data in the read buffer and asserting CYCEND. This deasserts the RdBusy signal. RdBusy is sampled in phase 1 by the processor, and it generates XEn in the next clock cycle. Since the one-deep read and write buffers are implemented using the same buffers, during a memory read operation the WrBusy signal is asserted to halt any write operations. The address enables are asserted through out the read operation to capture the addresses. For read operations, DAEN is asserted with MemRd signal to capture the data coming from the memory. The port B output enable for the data buffers is controlled by XEn for reading in the data. The read latency is five clock cycles including the fixup cycle.

For write operations in Figure 6 , the WrBusy signal is asserted as long as the write operation is in progress. This is indicated by WIP. It should be noticed that the RdBusy signal is asserted during write operations to block any read operations. The address enables are asserted during the write run cycle, and the address output enable is asserted throughout the write operation. The data buffer enables are asserted in the same way. It should be noted that WIP is a clocked output. The write operation takes three cycles.

Figure 7 shows a four word data block refill. The RdBusy signal, once deasserted, remains deasserted until MemRd is deasserted.

Figure 8 shows four word instruction block refill with streaming enabled. The instruction cache miss occured on the instruction I1. The refill starts with the basic block boundary instruction I0. The processor enters fixup as the missed instruction is fetched. The processor streams through the rest of the block.

Figure 9 shows a memory read requested by the processor before a previous write is retired to the main memory. The state machine puts out a request for the read operation only after the completion of the write operation, indicated by the first assertion of CYCEND. The enable ABEN is not enabled for the read untill the previous write is completed.

Figure 10 shows two write operations in two consecutive clock cycles. Since the write buffer is one word deep, the second write is not absorbed by the write buffer, and the processor stalls untill the first write is retired to the main memory. In the following fixup cycle, the second write is completed to the write buffer. The memory request MREQ for the second write is only generated in the fixup cycle. The data and the address of the second write are not captured by the buffers while the first write is in progress. In should be noted that deassertion of WrBusy is sampled by the processor in phase 1.

Figure 11 shows a write operation occuring in the middle of streaming. Streaming starts with instruction I1. The next instruction I2 issues a write. Since the write busy signal is already asserted, instruction streaming is aborted. The instruction I2 is executed in the following fixup cycle. WIP is asserted only in the fixup cycle. The data and the address of the write instruction I2 are not captured during streaming.

9

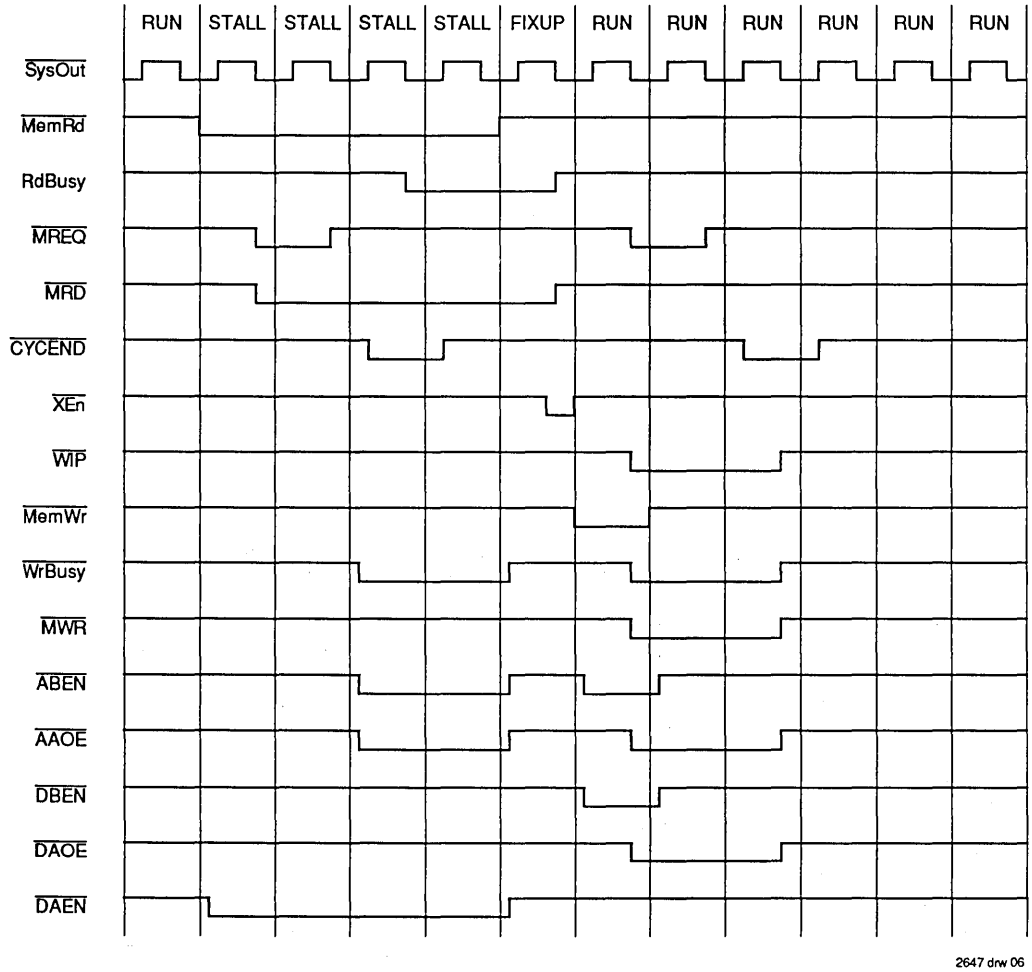


Figure 6. Read, Write Operations

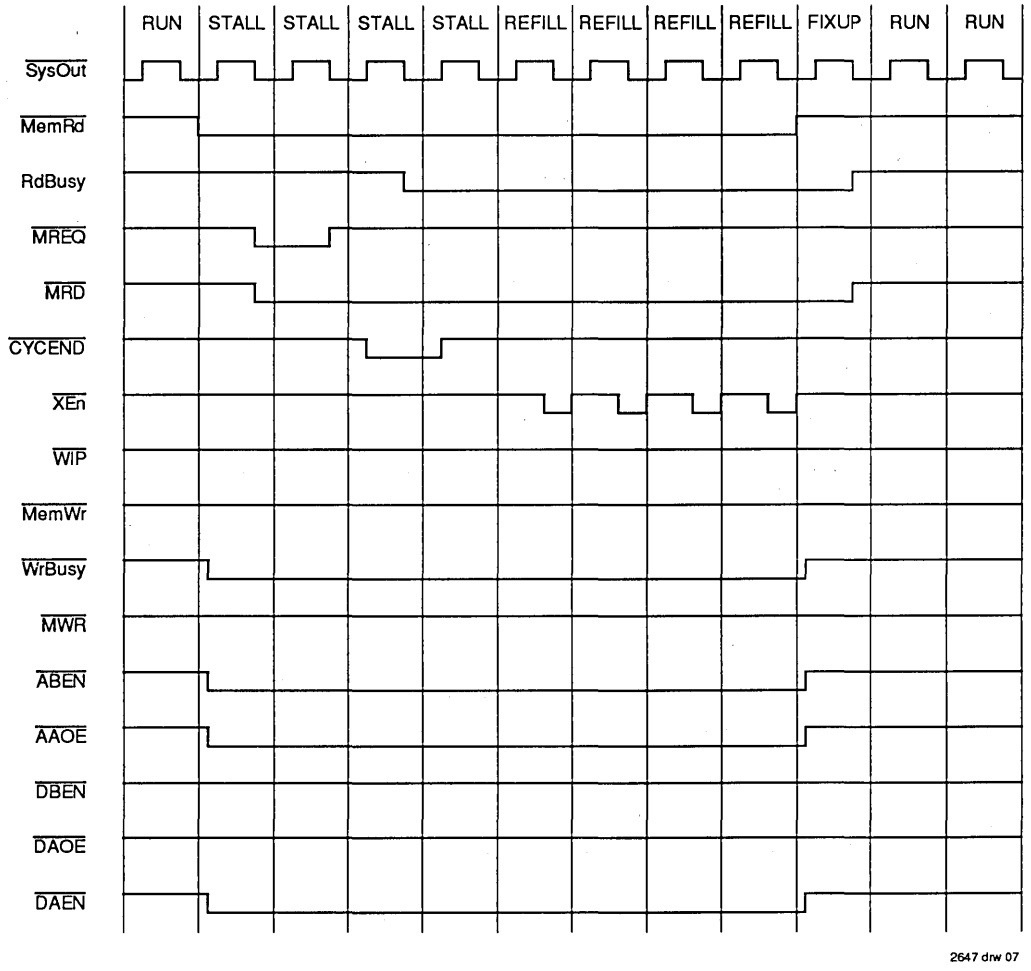


Figure 7. Data Block Refill

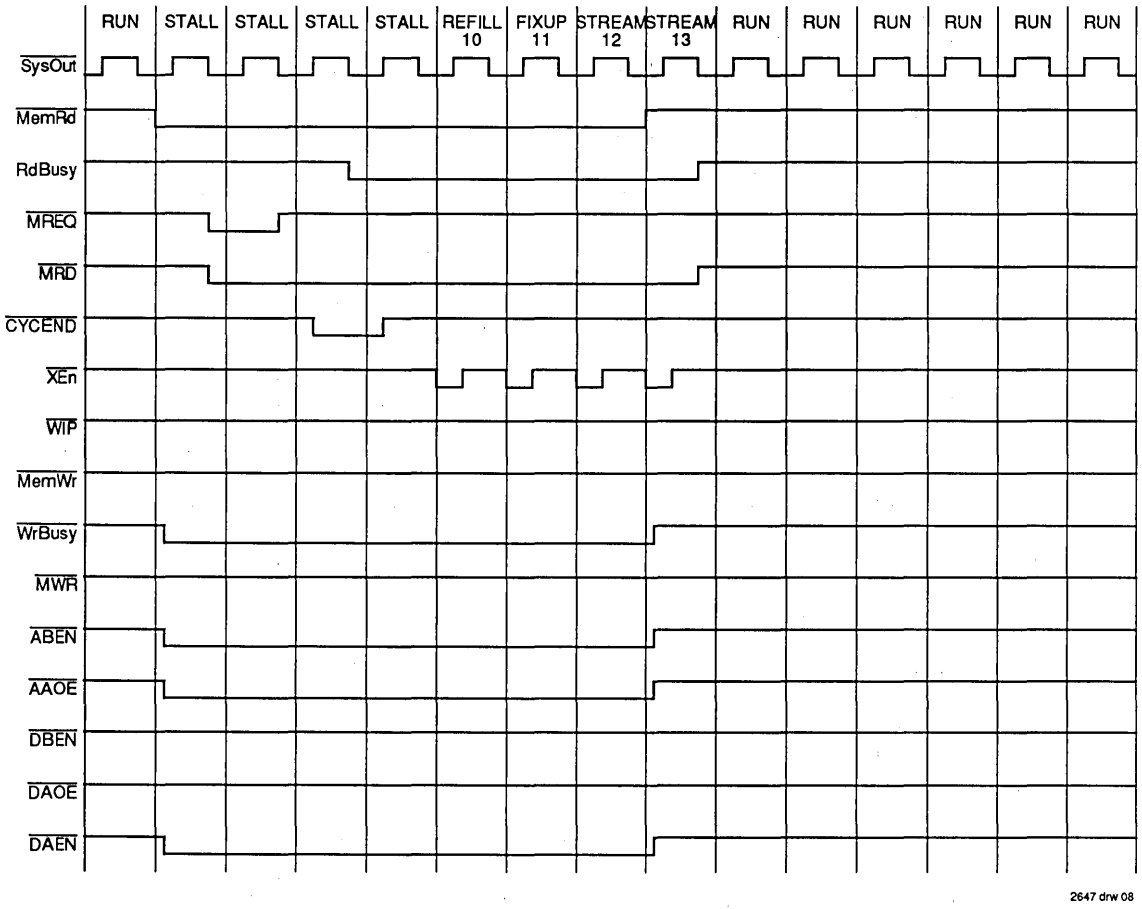
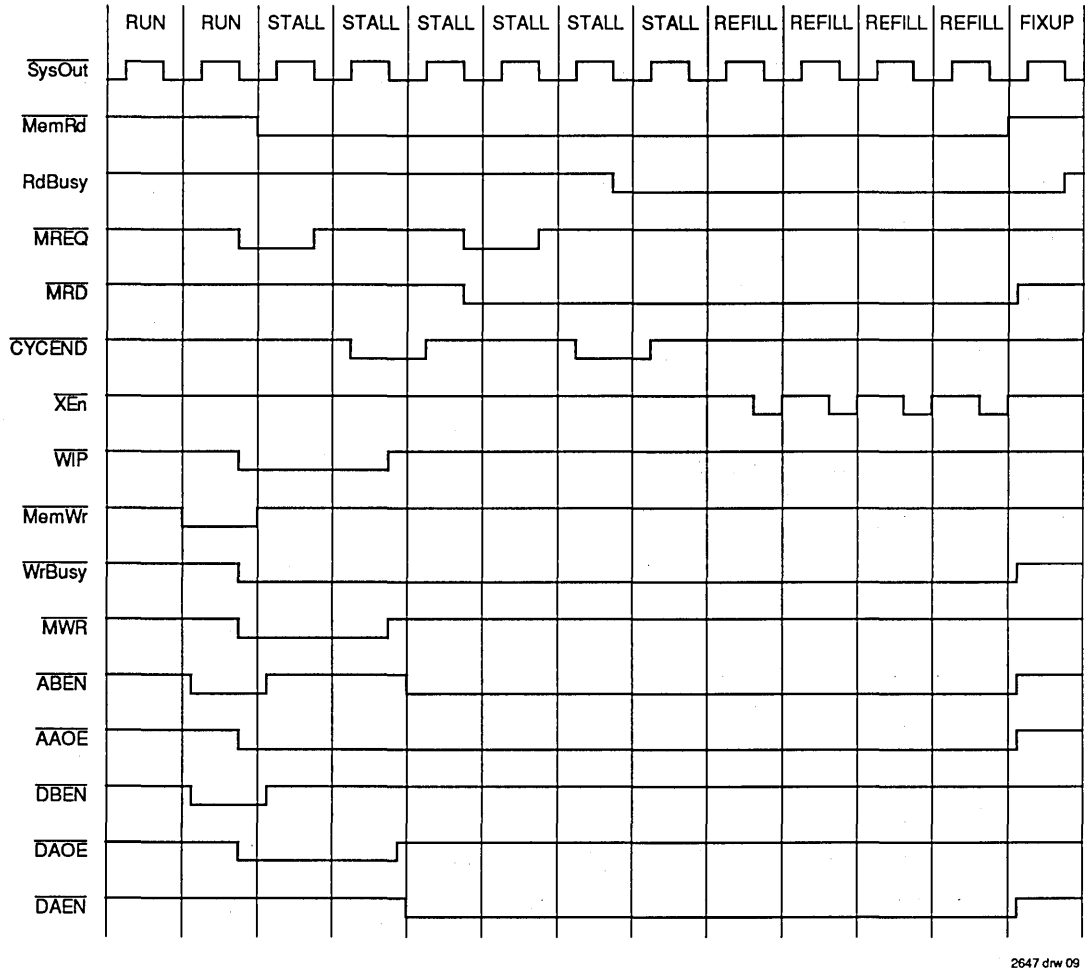


Figure 8. Instruction Streaming



2647 drw 09

Figure 9. Read During Write In Progress

9

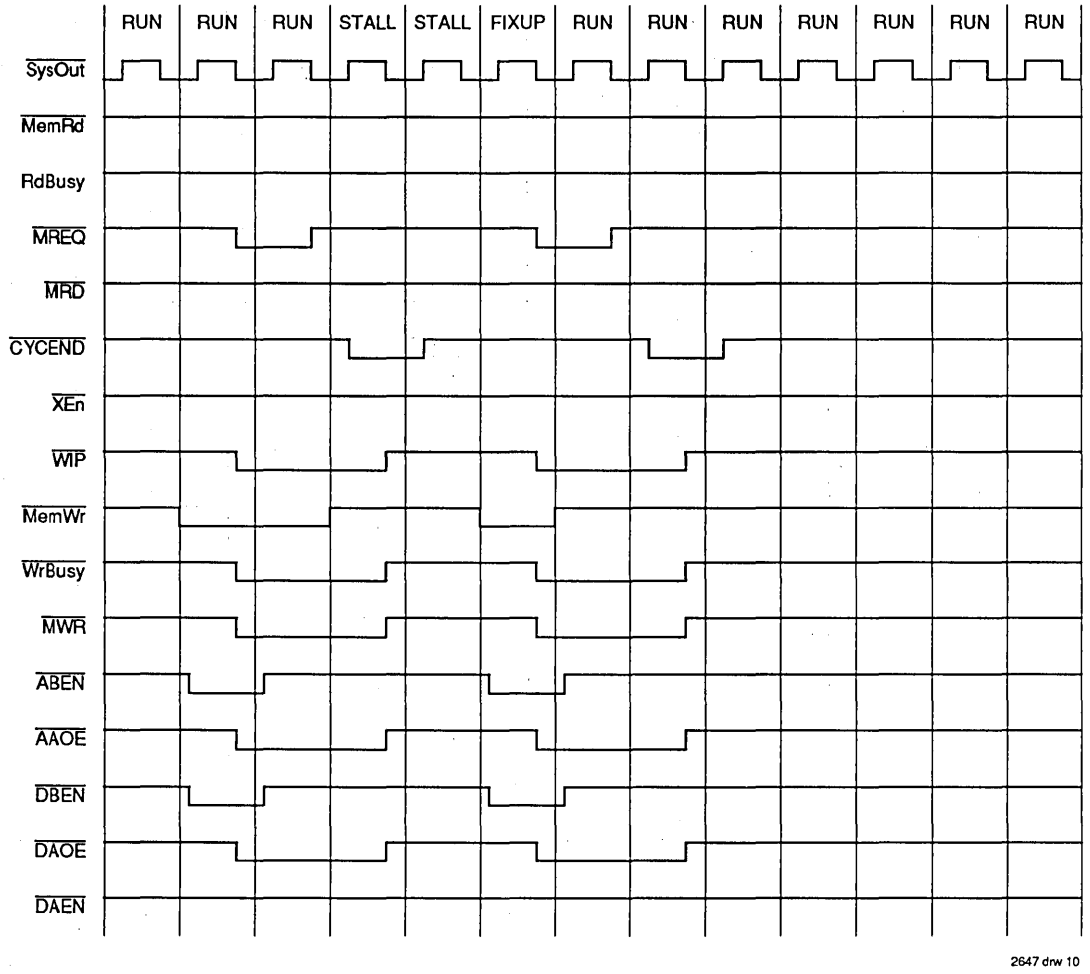


Figure 10. Write During Write in Progress

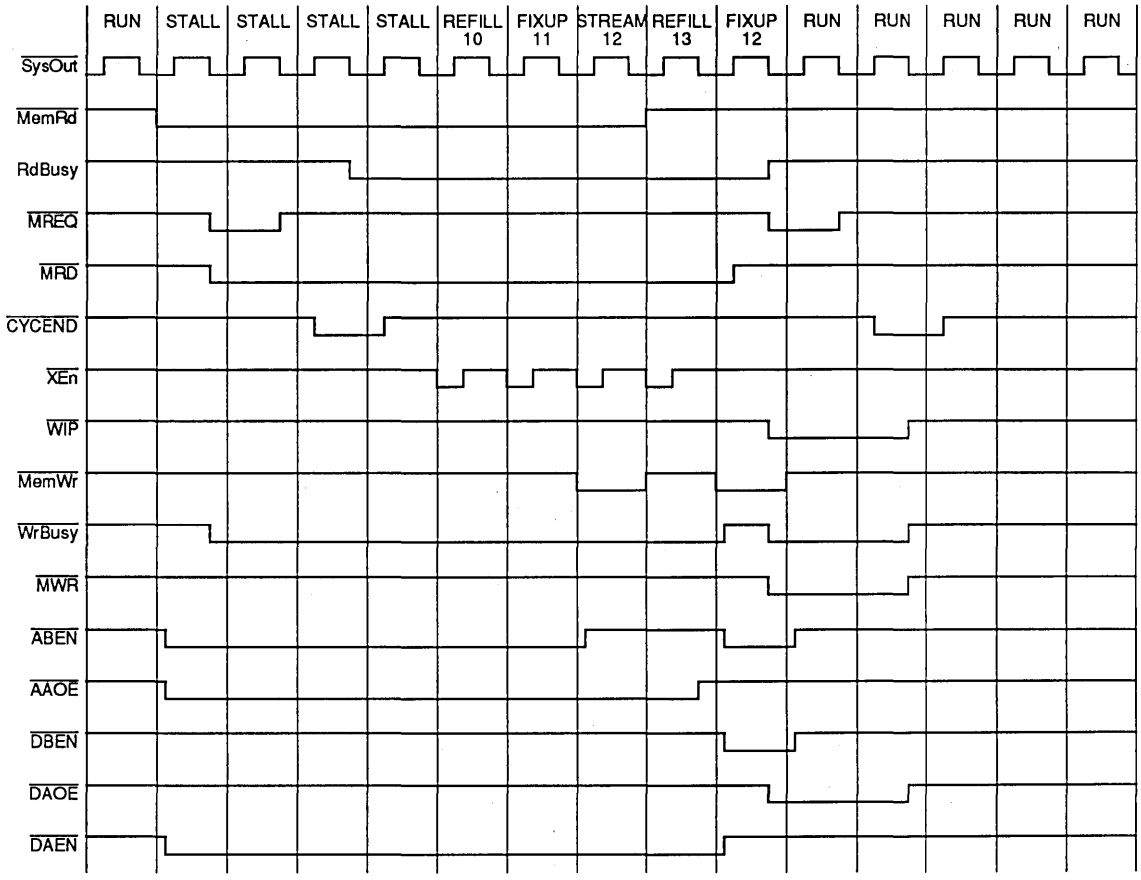


Figure 11. Write in Streaming

TWO DEEP READ AND ONE DEEP WRITE

IDT73210s can also be used in two-deep read and one deep write configuration. For capturing the addresses and the access type bits, four IDT73210s are used with B ports connected to the processor. For transfering data, four IDT73210s are used with A ports connected to the processor.

This configuration of the data path uses the registers Y and Z for read operations as two-level deep buffers. For write operations, the data is wrtten to the register X, thus providing a one-deep write buffer. The read and write data paths are shown in Figures 12 and 13. It should be noticed that even parity is generated on the data in both the directions.

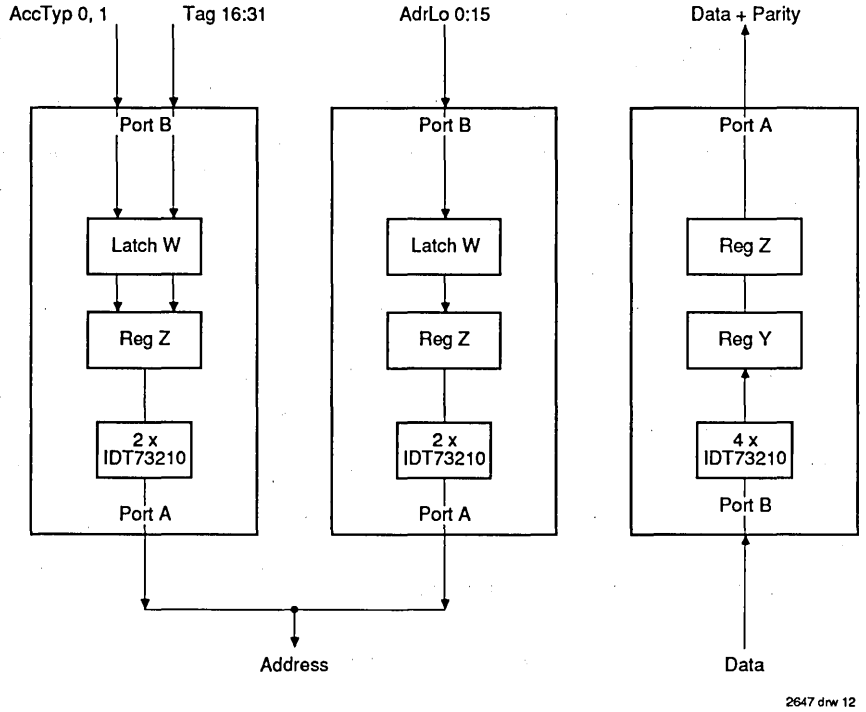
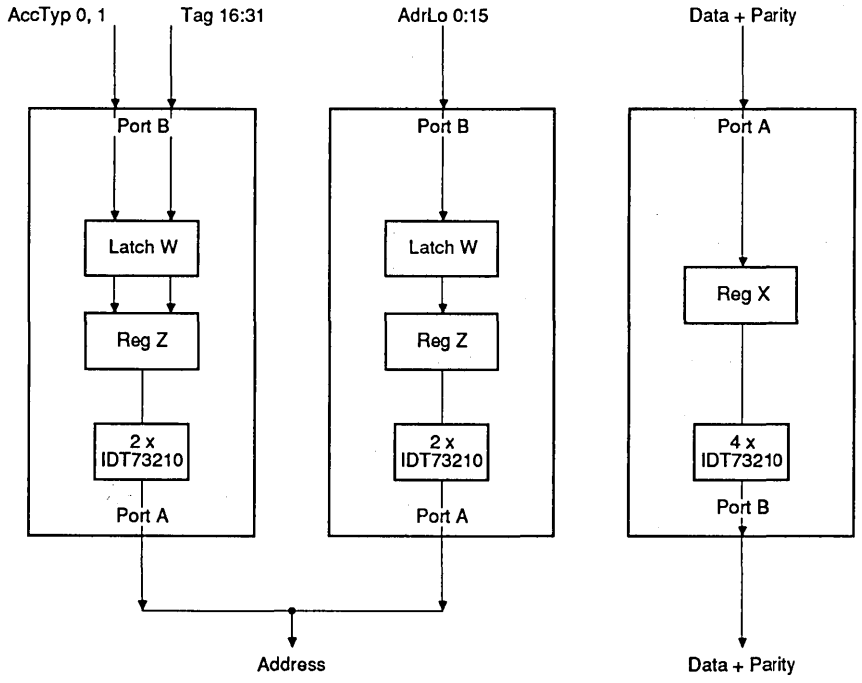


Figure 12. Read Data Path — Two Deep Read, One Deep Write Buffers Using IDT73210



2647 drw 13

Figure 13. Write Data Path — Two Deep Read, One Deep Write Buffers Using IDT73210

9

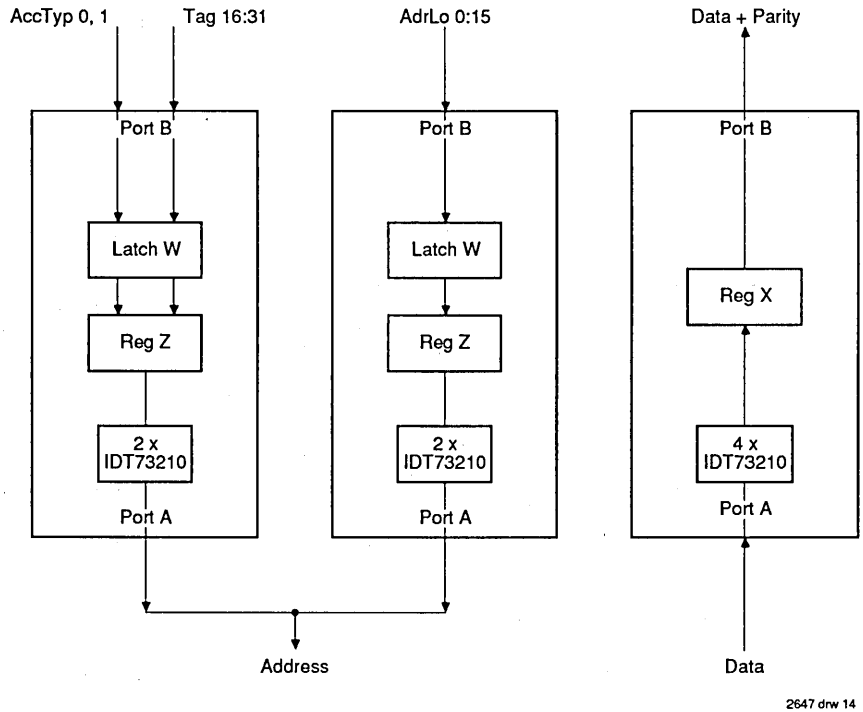


Figure 14. Read Data Path — One Deep Read, Two Deep Write Buffers Using IDT73210

ONE DEEP READ AND TWO DEEP WRITE

To use IDT73210s in a one deep read and two deep write configuration, four IDT73210s are connected to the address bus with B ports on the processor side. Four IDT73210s are connected to the data bus with B ports on the processor side to transmit data. The data path for the read operations is shown in Figure 14. The address and the access type bits can be passed through the latch W and the register Z. Data is read back from the memory through the register X after even parity is generated.

Figure 15 shows the write data path. To utilize the 73210s as two deep write buffer, the addresses and the data are passed through registers Y and Z. These two registers provide the two-level deep buffering for the addresses and the data. If any write operations, such as writing to the registers of I/O devices, require only one-deep write buffer, then the path through the latch W and the register Z is useful for both data and the addresses. It should be noticed that to transfer access type bits in two-deep write configuration, separate two-level deep buffering is required. Increasing the depth of the write buffer to two may improve the performance significantly if the application executes the second store before the first store is absorbed by the main memory.

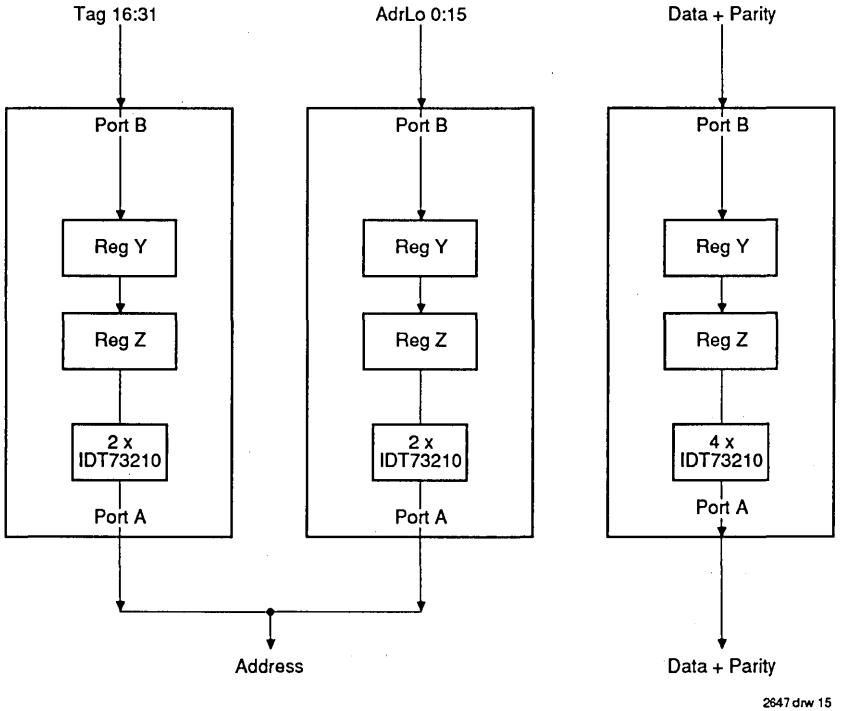


Figure 15. Write Data Path. One Deep Read, Two Deep Write Buffers Using IDT73210

CONCLUSIONS

IDT73210 is an ideal part for one/two-deep read/write buffers for R3000 applications. It is bidirectional, and speed compatible with the existing RISC processors. It generates and checks even parity and hence reduces the parts count in the memory interface for R3000 based systems. Using IDT73210s on the address bus, separate latches for capturing the address low bits can be eliminated. IDT73210 also provides the designer two different data paths from port B to port A to be selected dynamically depending on the operation.

9

Integrated Device Technology, Inc.

DESIGNING EMBEDDED CONTROL APPLICATIONS WITH THE IDT79R3001 RISController™

APPLICATION NOTE AN–66

By Michael J. Miller

INTRODUCTION

The IDT79R3000 RISC Microprocessor is increasingly selected for demanding embedded tasks in applications ranging from avionics controllers (CAP-32) to laser printer engines and PBX systems where performance is critical. This wide range of use is possible because of the inherently flexible yet powerful nature of the R3000 which allows the system designer maximum flexibility to achieve both cost and performance goals. The IDT79R3001 RISController is the first derivative of the R3000 family and builds upon this flexibility by allowing the designer to create systems with fewer cache parts or even systems without cache at all! While some processors incorporate an on-chip cache, it is typically smaller than desired for large applications. The R3001 allows the designer to decide whether to have a cache and also the cache size. Since the R3001 includes the cache control, there is no overhead beyond the actual memory components.

Basic system designs vary according to specific needs of the application. To be an ideal embedded control CPU, a processor must provide solutions that meet different sets of criteria. These criteria, however, vary in importance depending upon the particular application. In general, the criteria will fall into one of two broad categories: system performance or system cost. Performance can be measured in terms of raw throughput, context switch time and minimum and maximum latency to interrupts. System cost consists of component cost and parts count, which are not always the same. For avionics, system board space is paramount, not individual component cost. The IDT79R3001 RISController was defined to meet these requirements for embedded systems, as well as maintain software and architectural compatibility with the R3000. Compatibility allows all of the R3000 development tools to be used with the R3001. While the changes to the R3000 may seem deceptively simple, they result in significant benefits to the R3000 family. The 79R3001 dramatically reduces cache memory costs and lowers the interface parts counts, as well as provides support for single-hierarchy memory systems to increase performance.

TWO SYSTEM APPROACHES

The RISController offers the designer two basic system philosophies. The first is to treat a relatively small amount of local synchronous memory built with SRAM as a cache to allow the R3001 to manage the movement of data between the cache and main memory. In this configuration, the local memory provides each word of instruction or data with a valid bit and an address tag of its page in main memory. If the fetched data is not valid or is from the wrong place, the processor refetches the data from main memory in an asynchronous handshake. The designer programs the TAG address width to exactly match the system requirements. For example, a cacheable main memory of 8Mbytes and a cache of 16Kbytes requires a 9-bit tag and, thus, the total required width is 42 bits for data, tag and valid bit. With the large index address in the local cache, the designer can choose to lock into the cache portions of the program

The other approach is to incorporate a large amount of local memory and manage the loading of data and instructions through program control. With this approach, the width of the local memory is 32 or 36 bits, depending on whether or not parity is selected for protection. The R3001 accomplishes both approaches with direct control of the local memory through dedicated pins and, thus, avoids extra components and maximizes performance of the local memory.

RISController is a trademark of Integrated Device Technology, Inc.
C EXECUTIVE is a registered trademark of JMI Software Consultants, Inc.

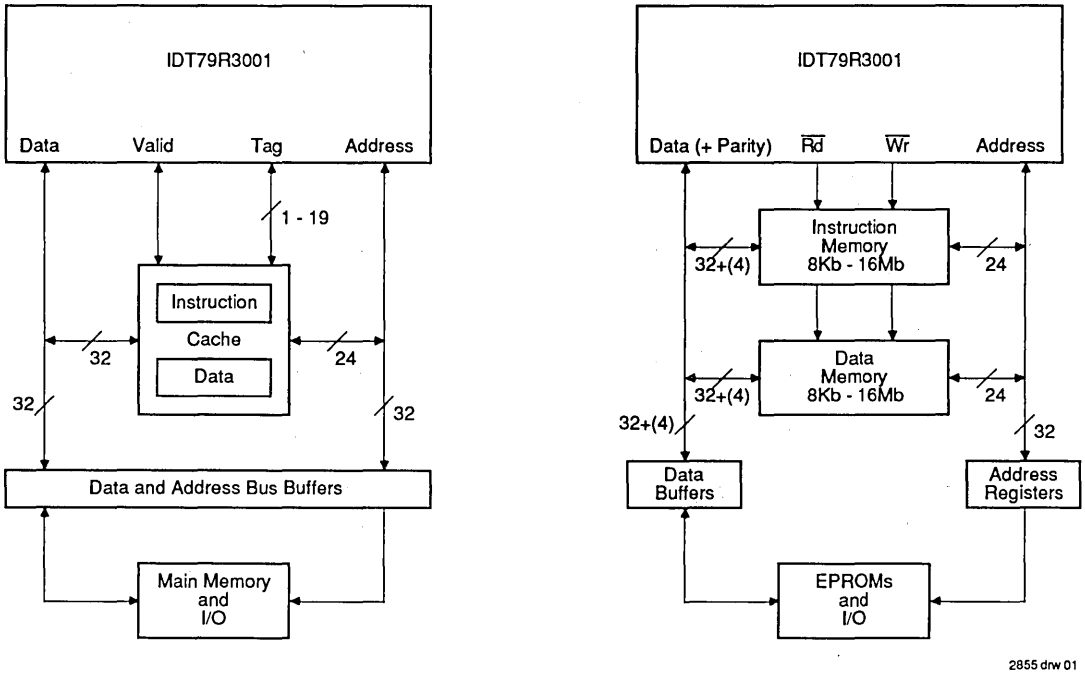


Figure 1. Block Diagram of Two Approaches

MAKING A SELECTION

To select the optimal configuration, the designer must determine the computational requirements of the application. For example, many applications have a "drop dead" performance requirement; a processor unable to achieve this minimum performance standard is not a viable candidate for the application. Performance can be measured as the time to execute a task, respond to an interrupt or perform a complete context switch. The performance of the R3000 is demonstrated to be 21 VUPS in a 25MHz M/2000 system or 13 VUPs in a 16MHz M/120 system. Using the JMI C EXECUTIVE® OS kernel on an M/120 provides context switches in 10μs and system calls in only 8μs.

Predictable interrupt response times can be achieved by locking the kernel into cache. With the R3001, this is accomplished by using a high address line (such as AddrLo23) as one of the address lines for the instruction cache. In this way the instruction cache is divided into two halves, each half containing words from a specific region in memory. In this example, if all of the kernel code is placed in memory above 8M and the application code is below 8M, the kernel will not be pushed out of the cache by the application code. Furthermore, if a kernel such as the "C EXECUTIVE" is used, the kernel will never leave the cache because it consists of only 8Kbytes which fit entirely into the 16Mb cache that is divided into two parts.

Another key aspect of an embedded control application is the parts count and the board area of the design. With this in mind, IDT and other SRAM suppliers currently provide wider memories as well as integrated logic functions on-chip. For example, the IDT71586 is a 4K x 16 SRAM with an integrated address latch that can be used to build a processor cache (for either the R3000 or 80386) with a low-parts count. Another new standard architecture is emerging with IDT's new 71222, which includes two interleaved banks of 4K x 18 SRAM in one device. With this approach, a designer can build a system that includes both instruction and data caches with minimal parts count. Like most other processors, the R3001 needs address and data buffers. If operated without cache it will provide performance similar to a 68020. By adding a 16Kbyte cache using only three parts (IDT71586), however, the performance more than doubles, making a 12.5MHz system achieve 4 VUPs running large embedded applications such as Page Description Interpreters.

9

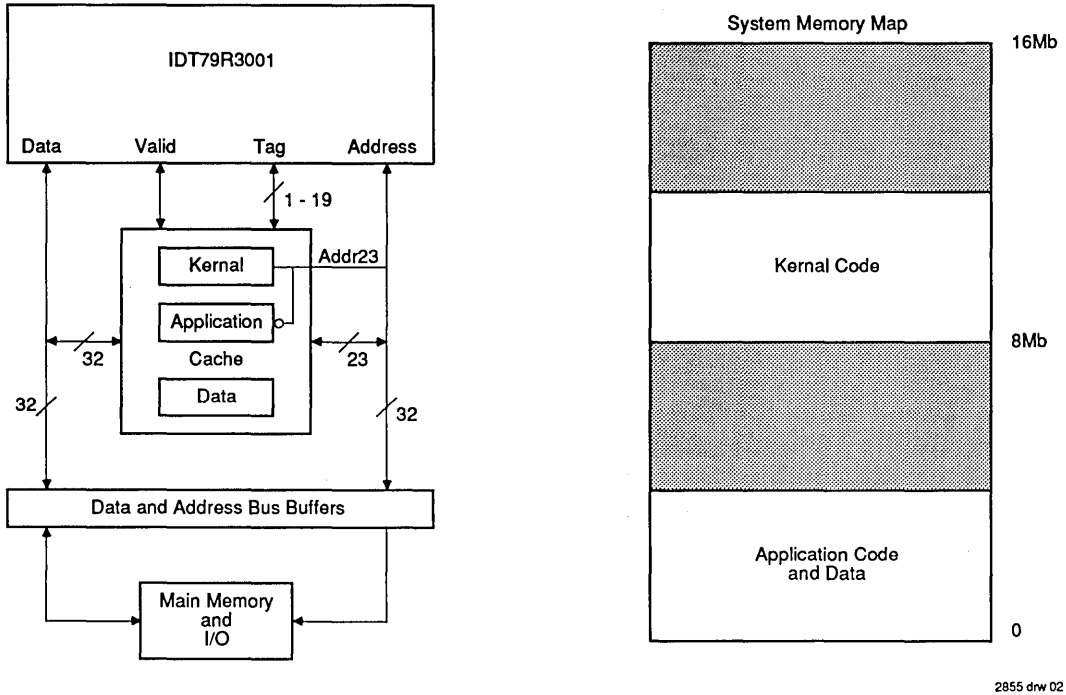


Figure 2. Locking Code in the Instruction Cache

NO CACHES

A "cacheless" scheme can be used in various embedded applications where deterministic behavior must be guaranteed, the application has stringent board space constraints and where the amount of code to be executed is limited although high-performance is demanded. Examples of such applications include communication controllers, avionics, robotics and switching controllers.

These systems typically feature compact instruction and data spaces because the nature of their task is highly repetitive. This allows the system designer to compact the memory hierarchy and eliminate the need to "cache" large main memory in his system. Instead, the R3001 is used in a non-hierarchical (without cache) based system, where all of the needed code and data fit in a tightly coupled synchronous memory. Such a system achieves the highest performance possible while minimizing the part count. Since all memory accesses are synchronous (with no cache or TLB misses), the system achieves the true (theoretical) performance of the RISC engine. At 25MHz operation, the system will achieve 23 VUPs with complete predictability for interrupts.

The example shown in Figure 3 is a system which incorporates 128Kb each of instruction and data synchronous memory, a one deep read/write buffer interface to slower peripheral devices, 64Kb of EPROM for instructions and initialization data (configured in the asynchronous space) and a DUART. The instruction and data memory can also be expanded by using 64K x 4 or 128K x 8 memories to make systems with 256K and 512K bytes, respectively.

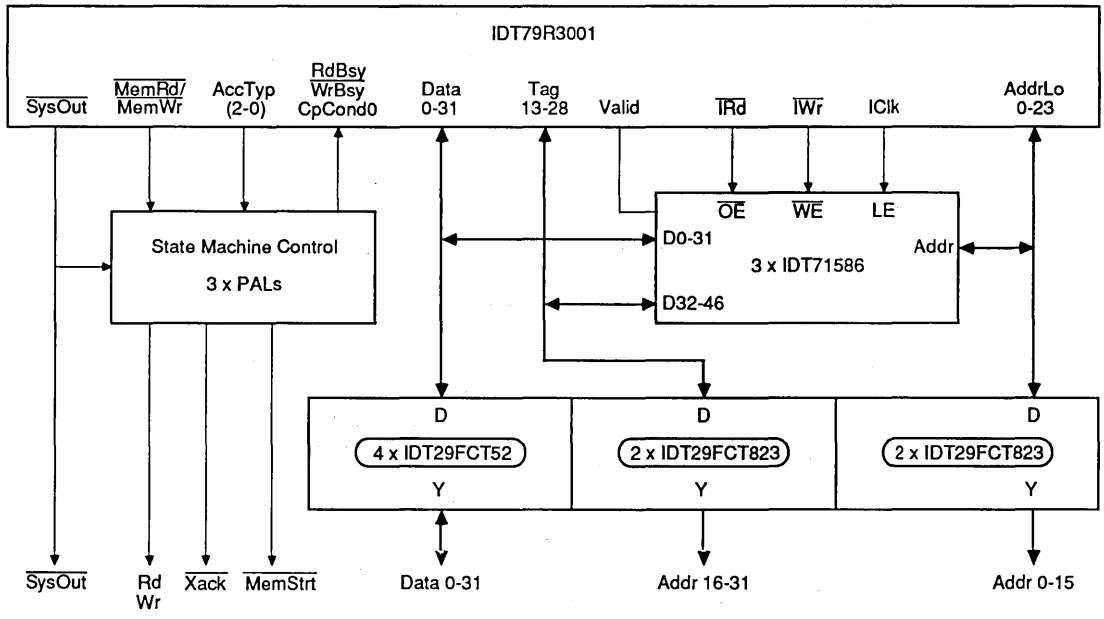


Figure 3. Low Parts Count System Using the IDT79R3001

Slower devices, such as the boot EPROM and the I/O devices, reside in the "asynchronous" address space of the system. The R3001 divides its 4GB virtual address space into 4 sections:

1. The kernel/user segment which is both cached and mapped by the TLB.
2. Kernel segment 0 which is not mapped but is cacheable. This is where the system's synchronous memory exists.
3. Kernel segment 1 which is neither mapped nor cacheable. The processor reset vector as well as the peripheral device are located in this space.
4. The kernel segment 2 which, like the kernel/user segment, is both cacheable and mapped by the TLB.

The asynchronous bus is used when accesses to kseg1 are issued by software, as well as when the system is initializing itself.

The asynchronous bus supports I/O and memory devices with varying access speeds. In this example, the asynchronous address space is 256Kb and the asynchronous data bus is 8 bits wide. The design includes 128Kb EPROM (two 64K x 8 device) and one DUART. The EPROM holds the boot code, initialization data and application code. It would be reasonably simple to scale this design to include other I/O devices as well.

CONCLUSION

One of the strengths of the R3000 is its ability to support different memory hierachies and cache configurations. IDT made modifications to the R3000 to create the IDT79R3001 RISController. It retains the flexibility of the original R3000 while meeting controller design requirements such as reduced parts count, small board real estate and predictable interrupt response. While future versions of the family will provide higher levels of integration such as on-board cache, there will always be a version which allows the designer to tailor the memory hierarchy.

9

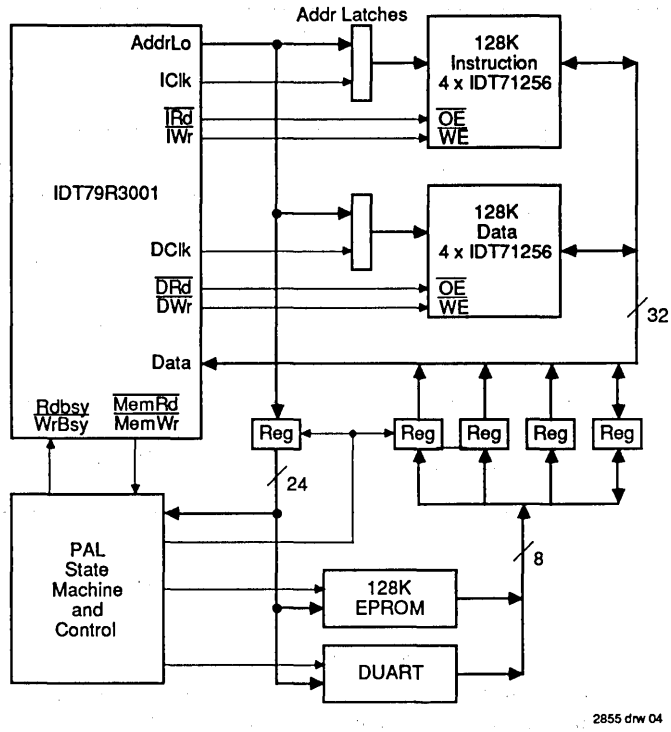


Figure 4. Using the IDT79R3001 in a Non-cache Approach

Integrated Device Technology, Inc.

USING IDT71502 RAMs IN A REAL-TIME DEBUGGING TOOL FOR A R3000 MICROPROCESSOR BASED SYSTEM

APPLICATION NOTE AN-67

by Bhanu V. R. Nanduri

INTRODUCTION

The proliferation of high-speed RISC and CISC microprocessors has created a demand for real-time debugging tools. This application note shows how a real-time logic tracing tool can be created using IDT71502 multifunction RAMs. The IDT71502 can be used as a stand-alone logic analyzer or as part of an embedded fault monitor and analysis system. Details of how to apply this system to an R3000 RISC microprocessor-based system are given. The discussion in this paper is also equally valid for use in high-speed CISC processor-based designs.

The IDT71502 can be used to function either as a logic tracing device or as a test pattern generator. As a logic tracing device the IDT71502 can record bus activity continuously and then be stopped on a predetermined event such as a bus error. This allows the activity leading up to the "event" to be recorded for analysis. Since the trace function is accommodated in a single device, embedded tracing is more likely to be practical.

DESCRIPTION OF IDT71502 MULTIFUNCTION RAM

IDT71502 is a 4K x 16 multifunction RAM with an address set-up time of 25ns. It has a breakpoint comparator, 16-bit pipeline register and an address counter. In addition, there is a 16-bit set-up register used to set the chip operating mode and to read back chip operating status conditions. It includes a serial control interface called the serial protocol channel (SPC™) which is available in a variety of other products from IDT as well. The SPC logic, as implemented in the IDT71502, has one 8-bit command shift register, a command decode register and a 16-bit data shift register. The serial data shift register can be configured to operate in a diagnostic mode. In the diagnostic mode of operation, the shift register can read all status conditions on the chip such as the RAM output, pipeline register output, data output pin state and RAM load/read counter value.

The serial protocol channel consists of a four-pin interface bus through which the user can access the internal registers of the IDT71502. The four pins are:

(a) Serial data input pin (SDI) for sending data and commands to the device.
(b) Serial data output pin (SDO) for extracting data from the device.
(c) Serial clock pin (SCLK) for clocking data and commands.
(d) Command/Data mode pin (C/$\overline{D}$) to provide command or data identification to the device.

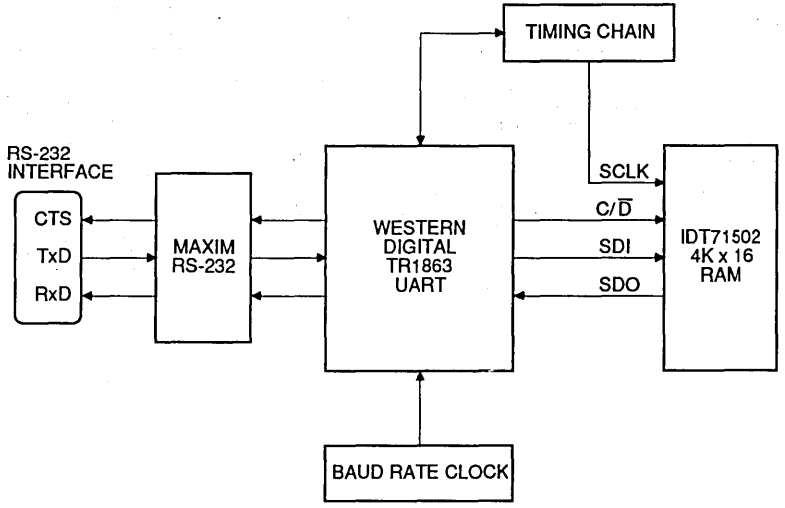


Figure 1.

SPC is a trademark of Integrated Device Technology, Inc.

9

This four-bit bus can be very conveniently connected to an RS-232C line for direct serial communication with a computer and Figure 1 illustrates one scheme to achieve this. The user is urged to refer to the "IDT71502 FUNCTIONALITY DEMONSTRATION BOARD USER MANUAL" for more information on interfacing the IDT71502 to the RS-232C serial communication line. The SPC's eight bit command is divided into a *four-bit command field* and a *four-bit register field*. The four-bit command field is used to determine whether a read or a write operation will be executed. The four-bit register field of the command register is used to select the various internal registers and the external pins on which the read or write will take place. Thus, the four-bit command field and the four-bit register field can effectively access any internal register for a read or a write operation and monitor the state of the external pins.

Table 1 summarizes the SPC commands, and Register codes and the set-up register format. When the command/$\overline{\text{data}}$ line is high, commands are serially clocked through the SCLK into the internal command register via the serial data input pin (SDI). When the command/$\overline{\text{data}}$ line is low, data is serially clocked by the SCLK into the internal data register via the serial data input pin (SDI). The SPC commands are executed whenever the C/$\overline{\text{D}}$ line transitions from a command mode (logic 1) to a data mode (logic 0). This device, when configured to operate in the trace mode, serves as a real-time debugging tool analogous to a logic analyzer.

SET-UP/STATUS REGISTER CODE

Bit	Name	Operation Performed
15	CE	Read Only
14	$\overline{\text{SOE}}$ FF	Read Only
13	$\overline{\text{SOE}}$ Pin	Read Only
12	$\overline{\text{OE}}$ Pin	Read Only
11	$\overline{\text{WE}}$ Pin	Read Only
10	$\overline{\text{INIT}}$ Pin	Read Only
9	BP Compare	Read Only
8	BP Pin	Read Only
7	$\overline{\text{CS1}}$	Read/Write
6	$\overline{\text{CS0}}$	Read/Write
5	Non-Reg High	Read/Write
4	Non-Reg Low	
3	—	—
2	BC-ADDRS	Read/Write
1	BC Pipelined	Read/Write
0	Trace Mode	Read/Write

2676 tbl 01

SPC COMMAND CODES

Command Code (Hexadecimal)	Read/Write Function	Operation
0	Read	Read Register
1	Write	Write Register
2	Read RAM	Read RAM and Increment Counter
3	Write RAM	Write RAM and Increment Counter
4-C	Reserved	(Reserved NO-OP)
D	Write	Stub Diagnostic
E	Write	Serial Diagnostic
F	Reserved	(Reserved NO-OP)

2676 tbl 02

SPC REGISTER CODES

Register Code (Hexadecimal)	Read/Write Function	Register
0	Read/Write	RAM Counter
1	Read/Write	RAM Output/Input
2	Read/Write	Pipeline Register
3	Read/Write	Break Mask Register
4	Read/Write	Break Data Register
5	Read/Write	Setup and Status Register
6	Read Only	I/O15 - I/O0 (Data Pins)
7	Read Only	RAM Address Pins
8-F	Reserved	Reserved (Unused)

2676 tbl 03

Table 1.

AN R3000-BASED SYSTEM

A block diagram of a R3000-based system's CPU and its memory interface is shown in Figure 2. It consists of the CPU and FPU, data and instruction caches and the read and write buffers connected to the CPU and the system bus. This is a typical configuration found in embedded or general purpose-type systems which use the R3000. To reduce the burden of the system designer who is interested in using the IDT79R3000, the CPU and FPU, as well as the instruction and data caches with the read and write buffers, are now available in a compact module (IDT7RS101) which can be connected to the user's system bus. This approach to system design vastly reduces the design cycle time by shifting the design emphasis to main memory and I/O interfaces.

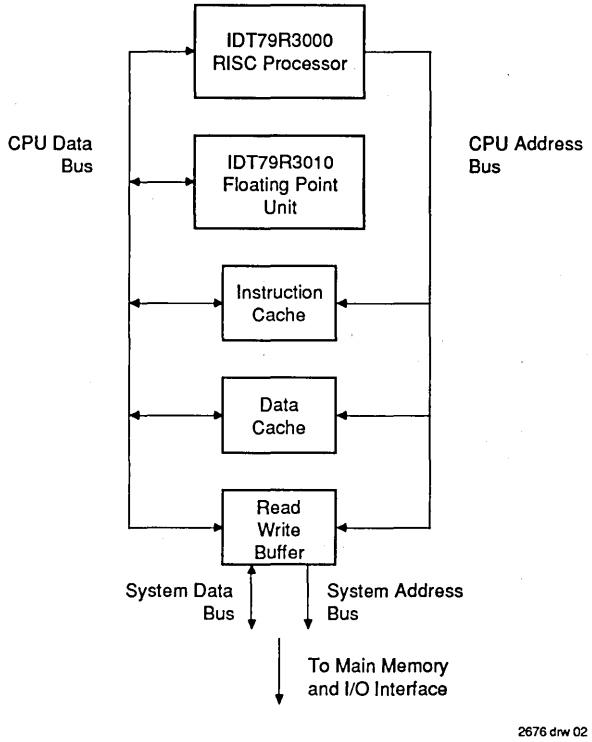


Figure 2. A Generic R3000 Microprocessor Based System

When debugging a system board based on the IDT7RS101 or its equivalent, the majority of debugging is done by monitoring the cache to main memory interface on the main memory side of this interface. An embedded trace function may operate in the same way. This keeps the capacitance of the trace RAM pins out of the speed critical cache buses. If desired, the R3000 can be operated in the uncached mode. This forces all accesses to main memory and allows every memory access of the processor to be monitored from the cache to main memory interface. The user wishing to operate in the uncached mode can do so by setting bit 11 of the TLB entry register to 1, indicating uncached mode, or operate the software in virtual address space kseg1. Kseg1 is kernal-mode virtual addressing space which is uncached and is 512 Mbytes long starting at virtual address 0xa000_0000. With this approach, the user must define instruction space and data space in the main memory and must provide an address decoded input to the IDT71502 tracing the control bus. This input will be used to determine whether an instruction or data related transaction occurred during that clock period.

Another approach is to tie the address valid bit on the TAG bus to ground via a 300 Ohm resistor. This is necessary to prevent a direct short from occurring when the CPU is driving the TAG bus. Tying the address valid bit on the TAG bus to ground will result in invalidating the cache TAGs and cache misses will occur, resulting in the processor accessing main memory to get that information. Whenever the main memory is accessed to get information after a cache miss, the processor puts out information on the Access Type pins, indicating the size of the word to be transferred and that it was a cached reference. AccTyp(2) pin output indicates a cached reference when 1 and an uncached reference when 0. AccTyp(0)

9

indicates a Data reference when 0 and an instruction reference when 1. The AccTyp signals are latched using our control trace RAM and will determine whether an instruction or data transaction occurred during that clock period.

A user wishing to implement his own cache can use the IDT71502 in the trace mode to monitor the cache. However, it should be pointed out that the timing for this part of his system is more stringent. The user may have to register trace data before clocking it into the IDT71502s to meet the IDT71502s set-up and hold time restrictions.

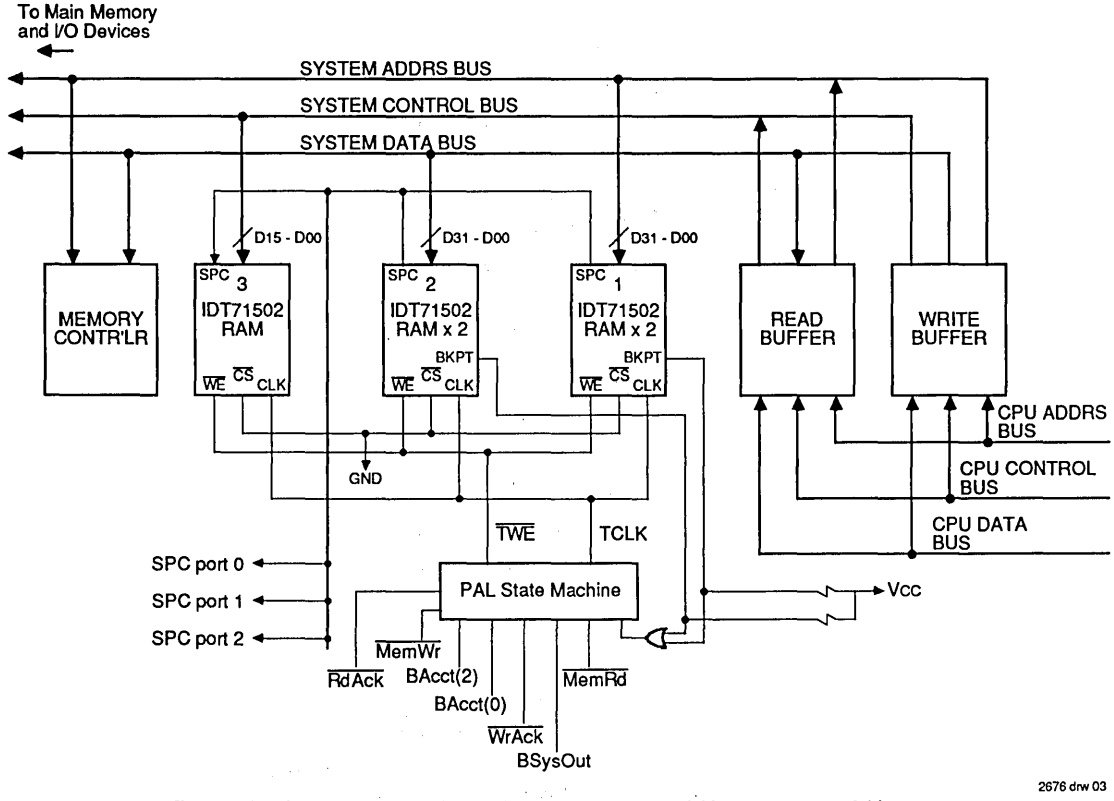


Figure 3. Block Diagram to Trace Instructions, Data, Instruction Addresses and Data Addresses on the System Bus of an R3000-based System

DESCRIPTION OF THE MONITOR CIRCUIT

Figure 3 shows the block diagram of an implementation of the monitor circuit. It is placed on the system bus between main memory and the write buffer of the R3000. The R3000 uses the write through cache update policy to ensure data coherency. The function of the write buffer is to capture data and addresses output by the CPU and ensure that data is passed on to main memory. The read buffer is used for temporary storage of data during data transfers between main memory and the CPU. Depending on the block refill size, the read buffer can be 1, 4, 8, 16 or 32 words deep. The block refill size of the system is fixed during the system reset operation. The R3000's CpCond0 input can be set to a 0 to indicate a single word transfer or can be set to a 1 to indicate a block transfer by the external memory controller. The PAL state machine is used to generate the appropriate IDT71502 strobes to capture instructions, data, instruction addresses and data addresses.

The IDT71502s labeled "1" in Figure 3 is used for capturing data and instruction addresses; IDT71502s labeled "2" is used for capturing data and instructions. The IDT71502 labeled "3" is used to trace the control bus signals. In this application note, we assume a single word deep read buffer. If a system is designed for all possible types of data transfers (i.e bytes, half words, tribytes, words and block refills), our PAL equations will also have to change to generate the strobes necessary to trace these data transfers.

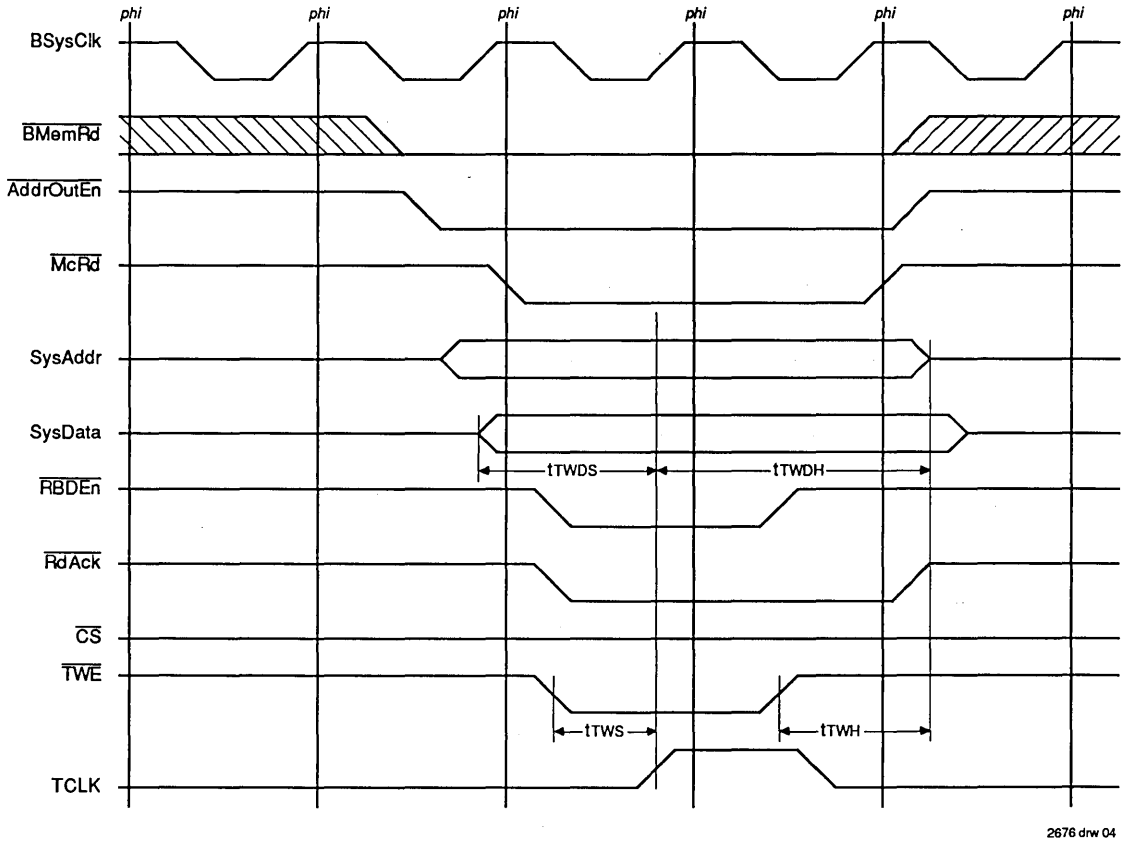


Figure 4. Main Memory Read Cycle (Single Word Read)

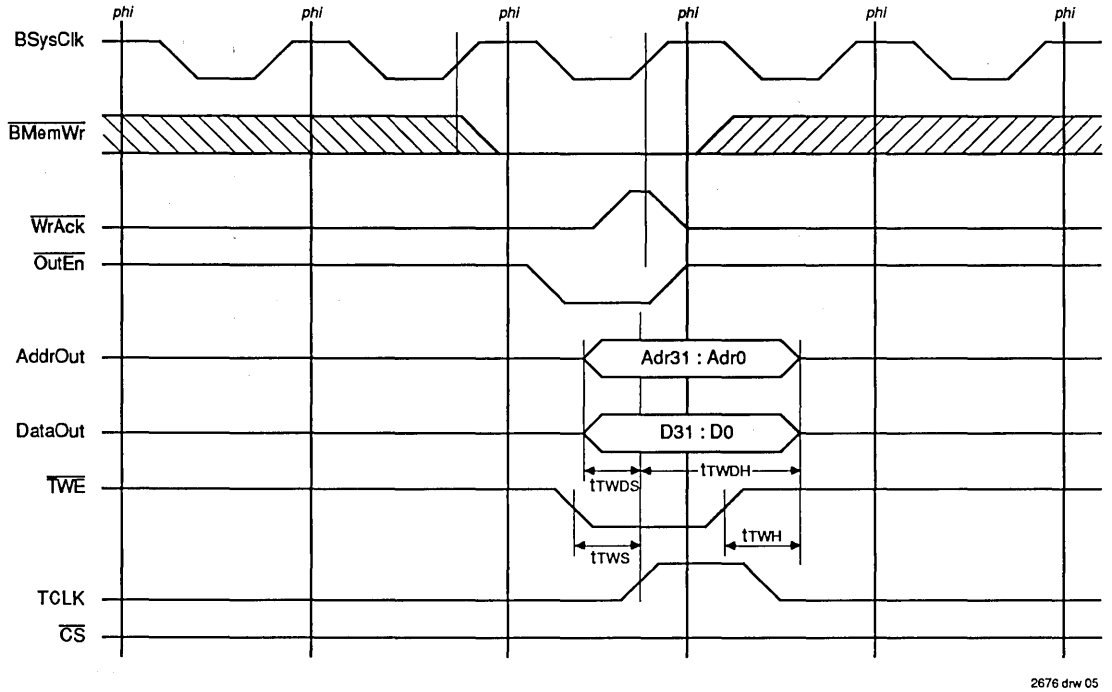


Figure 5. Main Memory Write Cycle (Single Word Write)

TIMING ANALYSIS

Figure 4 is the timing waveform for a single word read and Figure 5 is the timing waveform for a single word write. Since the system bus timing parameters are dependent on an external memory controller, Table 2 summarizes the important handshaking signals needed to satisfy the protocol necessary to trace system bus signals.

AddrOutEn is an input to the read buffer from the main memory controller. When asserted, this input will enable the address that is registered in the read buffer to the system bus. McRd is a read strobe that is generated by the main memory controller in response to a MemRd pulse from the R3000. RBDEn is a main memory controller input to the read buffer that registers the data available on the system data bus into the read buffer.

WrAck is an input to the write buffer from the main memory controller. It indicates that it has written the word presented to it to main memory. RdAck is also a main memory controller output that is used to generate the RdBusy signal to the R3000. TWE is an input to the IDT71502s that latches data addresses, instruction addresses, data, and instructions; it is also an output from our PAL state machine. TCLK is the clock input to the IDT71502s tracing data addresses, instruction addresses, data, and instructions. The signals TWE and TCLK are also used as inputs to the IDT71502s in order to trace the control bus signals.

Signal	Function
$\overline{\text{BMemRd}}$	The buffered memory read signal from the R3000
$\overline{\text{BMemWr}}$	The buffered memory write signal from the R3000
$\overline{\text{AddrOutEn}}$	Read buffer address output enable signal from the memory controller
$\overline{\text{McRd}}$	Main memory read strobe from the memory controller
$\overline{\text{RBDEn}}$	Read buffer data enable strobe from memory controller
$\overline{\text{WrAck}}$	Write acknowledge to write buffer from memory controller
$\overline{\text{RdAck}}$	Read acknowledge to read buffer from memory controller
$\overline{\text{TWE}}$	Write enable input to IDT71502 from PAL state machine
TCLK	Clock input to IDT71502 from PAL state machine

2676 tbl 04

Table 2.

TIMING SPECIFICATIONS FOR THE IDT71502s

t_{TWDS} is the IDT71502 specification defined as "Trace Write Data Set-up Time". The user must satisfy the following condition:

$$t_{TWDS} \geq 8ns$$

t_{TWDH} is the IDT71502 specification defined as "Trace Write Data Hold Time". The user must satisfy the following condition:

$$t_{TWDH} \geq 2ns$$

t_{TWS} is the IDT71502 specification defined as "Trace Write Enable Set-up Time." The user must satisfy the following condition:

$$t_{TWS} \geq 8ns$$

t_{TWH} is the IDT71502 specification defined as "Trace Write Enable Hold Time." The user must satisfy the following condition:

$$t_{TWH} \geq 2ns$$

CONCLUSION

The IDT71502 is a multifunction RAM that is fast enough to be used to trace the operation on most high-speed microprocessors including the IDT79R3000 RISC microprocessor. The 25ns speed grade can be used to trace full speed the operation of this processor up to 25MHz. The discussion in this paper focused on providing the pertinent information needed to construct a monitor circuit based on IDT71502 multifunction RAMs to trace the system bus of an R3000-based system. This discussion is also valid for users interested in using the IDT71502 RAMs in a trace mode to monitor system buses based on other high-speed processors.

Microprocessor based systems are usually provided with software routines that are used as diagnostic tools to test system primary and secondary memory for failures. These programs also test I/O devices before the user receives a prompt, telling him the system as a whole is ready for service. This procedure is usually carried out after system reset, but occasionally during normal operation the system "crashes" in the middle of some critical task and the user has no clue as to what happened prior to the "crash". The IDT71502 multifunction RAMs, when operated in trace mode and mounted permanently on critical system paths, can serve as "black boxes" to give the user this very important information. This information can then be very conveniently retrieved via the four bit serial protocol channel connected to the RS-232 connector and the reason for the crash can be determined.

The IDT71502 is a multifunction RAM that has the capability to serve as a valuable logic monitoring tool. It contains the Serial Protocol Channel and a breakpoint comparator, has a 4K x 16 memory space and is available with an access speed of 25ns. Thus, it is well-suited for use as a single chip logic analyzer in high-speed, high-density environs.

9

Integrated Device Technology, Inc.

USING THE IDT7MB6049 CACHE MODULE WITH THE IDT79R3000 RISC PROCESSOR IN SINGLE OR MULTIPROCESSOR SYSTEMS

APPLICATION NOTE AN-76

by Kelly Mass

The IDT7MB6049 is a complete cache module for the IDT79R3000 RISC processor and is designed for both single- and multi-processor systems. It has two banks of SRAMs, each configured as 16K x 60, and each with address latches. One bank is used to cache instructions, the other to cache data. They share a data bus, allowing one bank to be accessed at a time.

Use in multi-processor systems, is facilitated by a second address bus and an additional set of latches for that bus. This bus is used in multi-processor applications to latch an address from a source other than the R3000. This allows the system to invalidate entries in the data cache in conjunction with the R3000. This is done in order to maintain cache coherency. The set of address latches for the instruction cache is included in the module for symmetry, although normally no invalidations are done to the instruction cache. Instruction cache invalidation would require cache swapping, but only data cache invalidation is described below.

When the system wants to invalidate an entry in the data cache, it forces the R3000 into an MP Stall by asserting CpCond(3). During the one clock cycle that it takes for the processor to enter the MP Stall, it is the responsibility of the system to disable the output of the latch which supplies the processor's address to the data cache, and enable the output of the latch which supplies the invalidate address. The module pins $\overline{P1OE}(1)$ and $\overline{P2OE}(1)$ are used for this purpose. It is important that they should never be activated simultaneously since the outputs of the latches are tied together. The same applies to $\overline{P1OE}(2)$ and $\overline{P2OE}(2)$ for the instruction cache. Both address latches for the data cache are normally clocked by the same DClk signal from the R3000 through the P1LE(1) and P2LE(1) pins of the 7MB6049.

Once the processor is in MP Stall, it strobes $\overline{DRd}$ while CpCond(2) is unasserted, allowing the system to read the contents of the cache. The actual invalidation of the data cache entries begins when the system asserts CpCond(2) and provides the appropriate invalidate address. CpCond(2) causes the R3000 to output an invalid bit and strobe $\overline{DWr}$. Multiple invalidations are performed by keeping CpCond(2) and (3) asserted, and changing the invalidate address. Note that the invalidate address timing must be consistent with the processor timing. One suggestion is that the invalidate address input of the module be driven by a register that is clocked by SysOut.

The IDT7MB6049 has two chip select ($\overline{CS}$) signals. Both of these should be grounded if the cache is not depth expanded.

The four output enable ($\overline{OE}$) and four write enable ($\overline{WE}$) signals are split evenly between the data and instruction cache: (1-2) control the data cache, and (3-4) control the instruction cache.

$\overline{OE}(1\text{-}2)$ of the 7MB6049 connect to the $\overline{DRd1}$ and $\overline{DRd2}$ on the R3000. $\overline{DRd1}$ and $\overline{DRd2}$ are identical, and the load should be distributed evenly between them. Likewise, $\overline{OE}(3\text{-}4)$ connect to $\overline{IRd1}$ and $\overline{IRd2}$, $\overline{WE}(1\text{-}2)$ connect to $\overline{DWr1}$ and $\overline{DWr2}$, and $\overline{WE}(3\text{-}4)$ connect to $\overline{DWr1}$ and $\overline{DWr2}$.

The convention of the pin naming of the 7MB6049 is that P1 refers to the address from the R3000, and that P2 refers to the (invalidate) address from the system. Likewise, (1) refers to the data cache and (2) refers to the instruction cache. As shown in Figure 2, $\overline{P1LE}(1)$ and $\overline{P2LE}(1)$ are typically connected together to DClk since they latch addresses into the two data cache latches. $\overline{P1LE}(2)$ and $\overline{P2LE}(2)$ likewise are connected together to IClk. $\overline{P2LE}(2)$ is not used if instruction cache invalidation is not performed.

Similarly, $\overline{P1OE}(1)$ and $\overline{P1OE}(2)$ are typically connected together so that the outputs of the two R3000 address latches are enabled and disabled together, while $\overline{P2OE}(1)$ and $\overline{P2OE}(2)$ can together control the output of the invalidate address latches. $\overline{P2OE}(2)$ may be pulled continuously high if the instruction invalidate address latch is unused.

The 60 data I/O pins of the module are labeled D(0) to D(59). Although the ordering of the data and address pins of a RAM is normally arbitrary and can be ignored, that is not the case with the 7MB6049. Because of steps taken to reduce the chip count and power consumption of the module, Tag(12)-Tag(15) of the R3000 *must* connect to D(36)-D(39) on the 7MB6049, and AdrLo(12)-AdrLo(15) of the R3000 *must* connect to P1A(10)-P1A(13) on the 7MB6049. The order in which the other I/O pins are connected is not critical. Table 1 shows recommended I/O pin connections between the R3000 and 7MB6049.

R3000 Signals		IDT7MB6049 Signals
Data	Data(0) - Data(31)	D(0) - D(31)
Data Parity	DataP(0) - DataP(3)	D(32) - D(35)
Tag	Tag(12) - Tag(31)	D(36) - D(55)
Tag Parity	TAgP(0) - TagP(2)	D(56) - D(58)
Tag Valid	TagV	D(59)

2730 tbl 01

Table 1. Connection of Data and Tag Buses

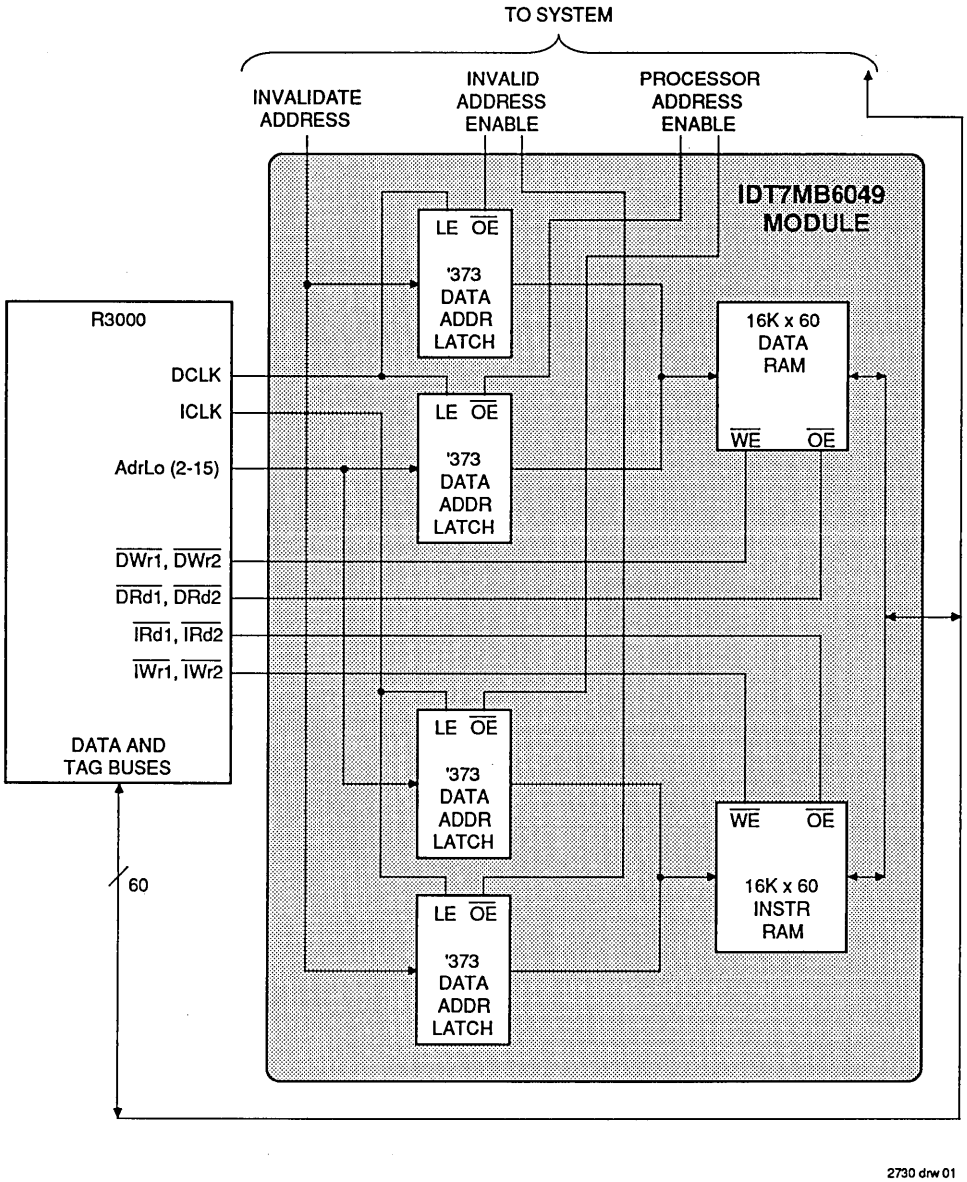
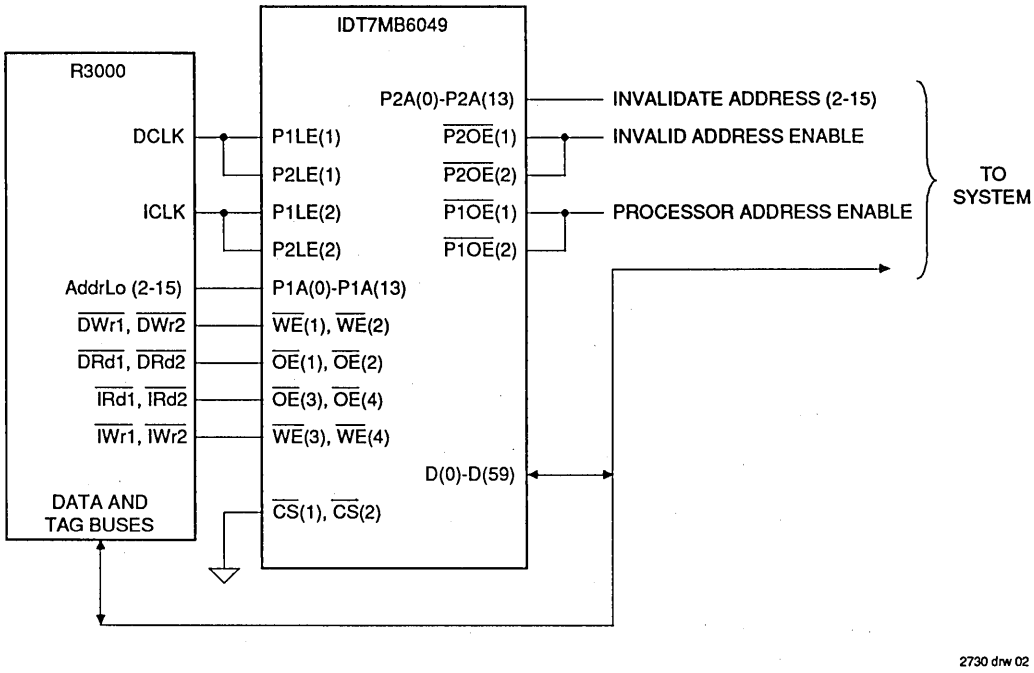


Figure 1. Block Diagram of the IDT7MB6049

9



2730 drw 02

Figure 2. Pin Connections of the IDT7MB6049

Integrated Device Technology, Inc.

IDT79R3001 SPECIFICATIONS & CACHE RAM TIMINGS

APPLICATION NOTE AN–77

1.0 GENERAL DESCRIPTION

The IDT79R3001 is a RISC microprocessor which is used in a variety of applications ranging from low-end embedded controllers to high-end workstations. Currently, the R3001 operates at a frequency of up to 33 MHz. This specification does not explain the functionality of the R3001 nor its architecture but is limited to describing the key timing parameters. For a more detailed description of the functionality and architectural description of the R3001, please refer to the references listed at the end of this document

This document starts with a brief description of the R3001, the three-phase clock inputs, cache timings, required timings for SRAMs to function as cache for the R3001, two technical notes explaining the factors used in the timing calculations, and the conclusion reached.

2.0 FUNCTIONAL DESCRIPTION

The IDT79R3001 is a 32-bit RISC microprocessor that is currently available from Integrated Device Technology, Inc. in two packages: the 144-pin PGA and the 172-pin ceramic flatpack. It has a 32-bit data bus, a 32-bit address bus that is divided into the low-order bits (AdrLo) and the high-order bits (TAG), control signals for the cache, control signals for main memory, and power and ground pins. The R3001 also has three double frequency clock inputs and one clock output used for interfacing the R3001 to the external world.

3.0 THREE CLOCKS AND DELAY-LINE SETTINGS

Figure 3.1 shows a block level diagram of the R3001 with its three clock inputs coming from a delay line. Table 3.1 shows a summary of the delay line settings to be used for different operating frequencies of the R3001. **Please note carefully that Clk2xSys is taken as the zero time reference and comes from the first tap of the delay line.** The other 2x clocks lag Clk2xSys in time and follow it with respect to delay line taps.

4.0 DERATING CALCULATIONS AND CACHE TIMING CONSIDERATIONS USING x4 SRAMS

The design of the cache subsystem for the R3001 is straightforward. Industry standard static RAMs function as cache. This chapter discusses the methodology used to calculate the critical timing parameters for a static RAM so that it can function as cache for the R3001. This chapter examines the timings for a 16 MHz, 20 MHz, 25 MHz, and 33 MHz R3001. The timing equations derived take into account the effect of capacitive loading on the bus. The derating factors are calculated based on certain assumptions. These assumptions are detailed in this chapter and the derating factors calculated. The timing equations are then discussed. At the end of this chapter a table containing the SRAM timings (for different operating frequency of the R3001) is included.

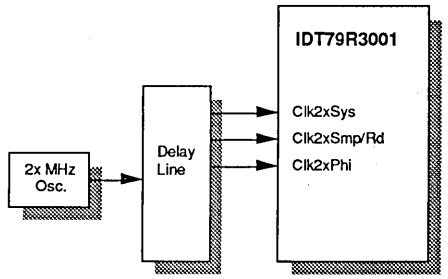


Figure 3.1. Three-Phase Clock Input to the R3001

Parameter	16 MHz	20 MHz	25 MHz	33 MHz
Clk2xSys	0	0	0	0
Clk2xSmp	6	6	6	4.5
Clk2xPhi	16	14	12	9

Table 3.1. Delay Line Settings for R3001 Operating at Different Frequencies

9

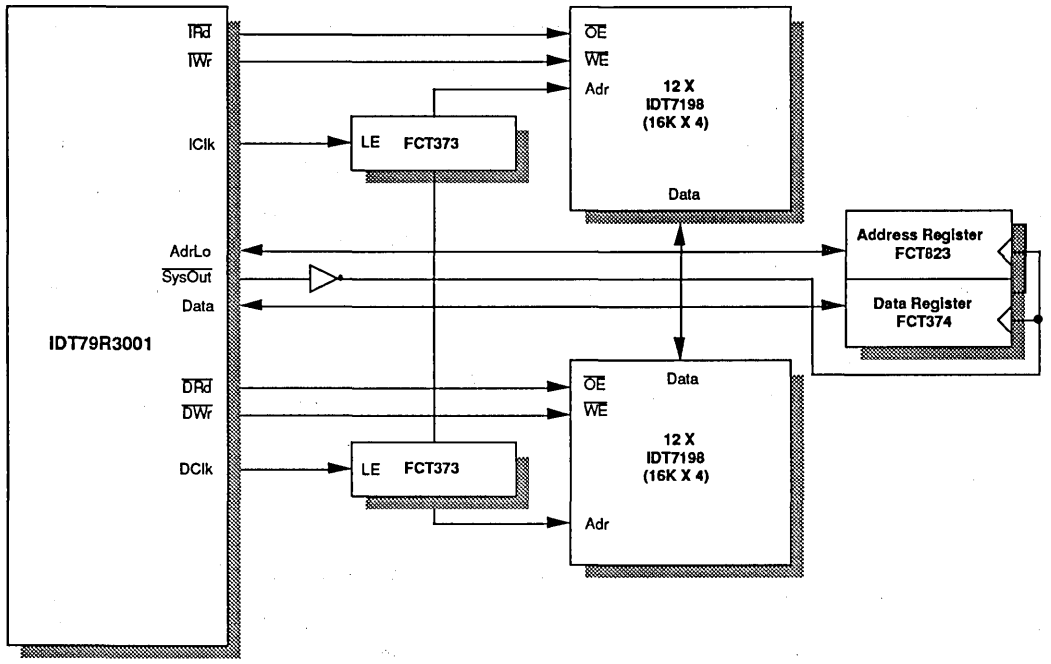


Figure 4.1. Block Level Diagram of a Cache Subsystem with the R3001 Using IDT7198 16K x 4 to Function as Cache (the R3010 is not shown in this figure)

4.1 Device Capacitance

Figure 4.1 shows a typical R3001 based system. The cache comprises of fast 16 K x 4 static RAMs e.g., the IDT7198. The AdrLo bus of the R3001 goes through a high-speed transparent latch: the FCT373. It also goes through a latch which is used to address the main memory. All the devices have an input and an output capacitance. In addition, each device is capable of driving a certain load. These parameters: the input capacitance, the output capacitance, and the load capacitance are given in Table 4.1.

The cache format of the R3001 comprises of 48 bits: 32 bits of data, 15 bits of tag and, a valid bit. With this requirement, it is clear that for the instruction cache, 12 IDT7198s (16K x 4 SRAMs) are needed. The data cache has the same format. This means that there are a total of 24 SRAM devices for the cache.

Device	# of Devices	Capacitance	Total Capacitance
R3000	1	Cin = 10 pF	10 pF
IDT7198	2	Cout = 7 pF	14 pF
IDT374A	1	Cout = 12 pF	12 pF
IDT823B	1	Cin = 10 pF	10 pF

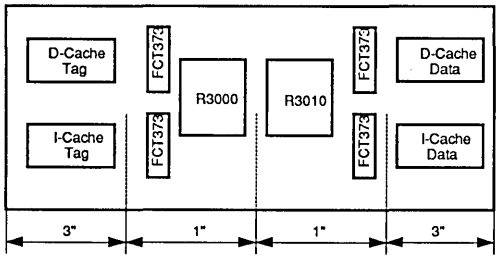
Table 4.1. Capacitances of the Various Devices in a Typical R3001 System

4.2 Assumptions for Surface Mount Layout Design With 'x4' SRAMs

In the following sections, certain assumptions have been made while calculating the derating factors. These are as follows:

1) The trace has a capacitance of 2 pF/inch.
2) The speed of light is 2 ns/foot in epoxy.
3) The R3001 speeds are specified with a loading of 25pF. For every additional 25 pF, there is a delay of 1 ns.
Note that the cache control signals are specified with a 50pF load and derate 1ns/25pF after that.
4 The distances between the R3001 and the latches are approximately 1 inch each.
5) The distances between the R3001 and the RAMs ap proximately 4 inches each.
6) In all of the assumption, it is assumed that a surface mount package is used.

Figure 4.2 shows a brief mechanical layout of an R3001 board.



Assume read and write buffer underneath the board

Figure 4.2. Surface Mount Board of an R3001 System with Cache and Main Memory Interface and Approximate Distances Between the Various Devices

4.2.1 Address Bus Derating Calculations

For the system shown in Figure 4.1 each address bit is connected to five latches: one going to the main memory interface buffer, two to the instruction cache and tag memory, and two to the data cache and tag memory respectively. The latches in turn are connected to the address pins on the static RAM. Figure 4.3 shows all the devices that each address bit is connected to.

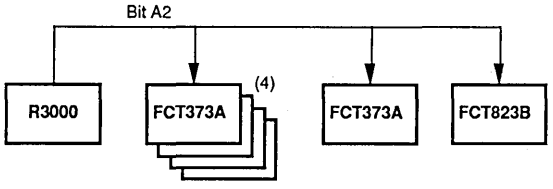


Figure 4.3. Block Diagram Showing Various Devices Connected to One Address Bit

Trace length from the CPU to the latch = 4 inches (4.1)
Capacitance of the trace, C_{trace} = 4 X 2 pF/inch = 8 pF (4.2)
Input capacitance of the 373 latch = 10 pF (4.3)
As each address bit is connected to five latches (FCT373),
Total input capacitance due to 5 devices,
C_{373in} = 5 X 10 = 50 pF (4.4)
Total capacitive load = C_{trace} + C_{373in} = 8 + 50 = 58 pF (4.5)
The rated R3001 load, $C_{L(R3001)}$ = 25 pF (4.6)
From Eq. (4.5) and Eq. (4.6), Extra capacitive loading for the R3001 = 58 - 25 = 33 pF (4.7)

Let us now examine the capacitive loading between the latches and the RAM.

Path length from latches (373s) to RAM (7198s) = 3" (4.8)
Trace capacitance from latch to RAM = 3 X 2 pF/in = 6pF (4.9)
Input capacitance of the RAM = 5 pF (4.10)

Each output from the latch is connected to eight RAM devices.
Load due to 8 devices = 6 X 5 = 30 pF (4.11)
Total capacitance = 30 + 6 = 36 pF (4.12)
The rated '373 load = 50 pF (4.13)

From Eq. (4.12) and Eq. (4.13) it can be seen that there is no delay due to the capacitive load between the latch and the RAM.

However, there is a delay due to the capacitive load between the R3001 and the latch. This delay can be calculated as follows:

For every extra 25 pF of load, there is a delay of 1ns (4.14)
From 4.7 and 4.14, delay due to the capacitive load = 33 / 25 = 1.32 ns (4.15)
The speed of light ≈ 2 ns/foot (4.16)
For a maximum path length of 5", delay = 5"/12" X 2 = 0.8 ns (4.17)
From Eq. (4.15) and Eq. (4.17),
Total propagation delay for the address bus, $AdrLo^d$ = 1.32 + 0.8 ≈ 2 ns (4.18)

4.2.2 Data Bus Derating Calculations

The derating calculations for the data path are similar to those done for the address path. The data bus is connected to the floating point unit (R3010), the instruction cache (IDT7198), the data cache (IDT7198), a read register (FCT374A), and a write register (FCT823). This is shown in Figure 4.4. Two cases must be considered: a data store and a data fetch. Both are discussed.

9

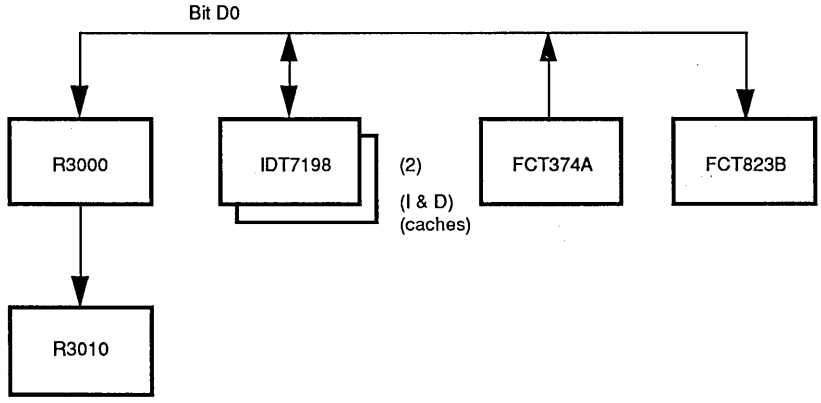


Figure 4.4. Block Diagram Showing Various Devices Connected toOne Data Bit

4.2.2.1 Data Store (R3001 CPU Outputs Data):

Each data bit is connected to two RAMs (7198s) — one for instruction and one for data.

The path length for the data bus = 5" (4.18)
Trace capacitance for the data bus = 5 X 2 pF/in=10pF (4.19)

Capacitive loading due to devices,
$C_{devices} = 2 \times C_{RAMin} + C_{374in} + C_{823} + C_{R3010}$ (4.20)
$C_{devices} = 2 \times 7 + 12 + 10 + 10 = 46$ pF (4.21)
$C_{total} = C_{trace} + C_{devices} = 46 + 10 = 56$ pF (4.22)

Propagation delay due to speed of light =
5"/12" x 2 = 0.8 ns (4.23)
Delay due to capacitive load = (56 -25) /25 = 1.24 ns (4.24)

From Eq. (4.23) and Eq. (4.24),
Total propagation delay on a store = 1.24 + 0.8 ≈ 2 ns (4.25)

4.2.2.2 Load (RAM Provides Data)

Since the trace length is the same, C_{trace} = 10 pF (4.26)
Capacitive load due to devices,
$C_{devices} = C_{R3001} + C_{R3010} + C_{RAMin} + C_{374in} + C_{823}$
$C_{devices} = 10 + 10 + 12 + 10 + 7 = 49$ pF (4.27)
Total capacitance = $C_{trace} + C_{devices} = 10 + 49 = 59$ pF
The RAM rated drive is 30 pF.
Extra load = Total capacitance - RAM rated drive =
59 - 30 = 29 pF (4.28)

Propagation delay due to capacitive load = 29/25 = 1.16 ns (4.29)
Propagation delay due to the path length = 0.8 ns (4.30)
Total propagation delay = 1.16 + 0.8 ≈ 2 ns (4.31)

4.2.3 Read and Write Control Derating Calculations

The effect of the capacitance on the control signals from the R3001 processor to the caches and the memory interface is considered here. The control signals on the R3001 are the $\overline{IRd}$, $\overline{DRd}$, $\overline{IWr}$, and $\overline{DWr}$ which control the instruction cache read, data cache read, instruction cache write, and data cache write respectively. The read and write control signals are connected to the output enable ($\overline{OE}$), and write enable ($\overline{WE}$) of the instruction and data cache, respectively. Assuming the use of a 16 K x 4 IDT7198 static RAM, each control signal is connected to 8 such static RAMs.

Number of devices (SRAM) connected to each control line = 12 (4.32)
Input capacitance of each device (SRAM) = 5 pF (4.33)
Total load capacitance = 5 x 12 = 60 pF (4.34)
Path length = 5" (4.35)
Trace Capacitance = 5 x 2 pF/in = 10 pF = 10 pF (4.36)
Total capacitance = 60+ 10 = 70 pF (4.37)
Extra capacitive load = 70 - 50 = 20 pF (4.38)
Propagation delay due to capacitive load ≈ 1 ns (4.39)
Propagation delay due to the trace length = 0.8 ns (4.40)
Total propagation delay = 1 + 0.8 ≈ 2 ns (4.41)

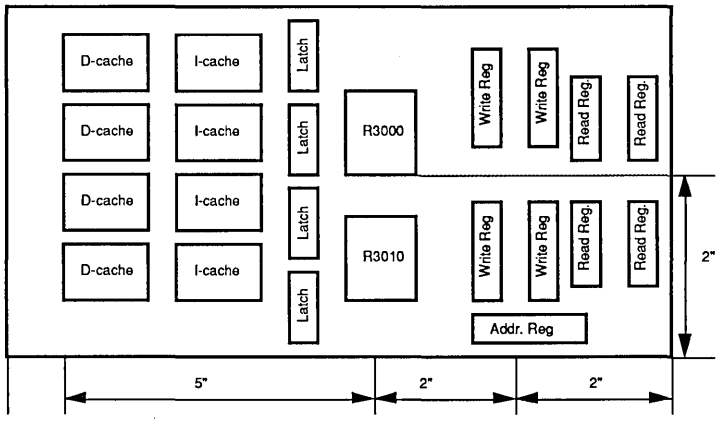


Figure 4.5. Board Layout for a Through-Hole Design of an R3001 Cache Subsystem

4.3 Assumptions for Through-Hole Layout Design Using x4 SRAMs

In this section, the deratings are calculated for a through hole design. Figure 4.5 shows an example of the layout of a through hole design. This layout corresponds to an demonstration board used extensively at IDT. The data trace lengths are 10 inches and the address trace lengths are 9 inches.

4.3.1 Address Derating Calculations

For the system shown in Figure 4.3, the number of devices connected to the R3001 is the same.

Trace length from the CPU to the latch = 9 (4.42)
Trace capacitance = C_{trace} = 9 x 2 = 18 pF (4.43)
Input capacitance of the latches = 10 pF (4.44)
Total capacitance = 5 x C_{373} + C_{trace} = 5 x 10 + 18 = 68 pF (4.45)
Extra load on the R3001 = C_L = 68 - 25 = 43 pF (4.46)
Since the rated 373 load is 50 pF, there is no derating factor between the FCT373 and the RAMs.
Therefore the derating is between the R3001 and the latches.
Delay due to capacitance = 43/25 = 1.75 ns (4.47)
Propagation delay due to the trace length = 9/12 x 2 = 1.5 ns (4.48)
Total derating on the address bus = 1.75 + 1.5 ≈ 3 ns (4.49)

4.3.2 Derating on the Data Bus

As in section 4.2.2, the derating for the data bus is calculated for two cases: i) an instruction fetch, and ii) data store.

4.3.2.1 Data Store (R3001 CPU Outputs Data)

Each data bit is connected to two RAMs (7198s) - one for instruction and one for data.
The path length for the data bus = 10" (4.50)
Trace capacitance for the data bus = 10 x 2 pF/in = 20 pF (4.51)
Capacitive loading due to devices,
$C_{devices}$ = 2 x C_{RAMin} + C_{374in} + C_{823} + C_{R3010} (4.52)
$C_{devices}$ = 2 x 7 + 12 + 10 + 10 = 46 pF (4.53)
Total capacitive load = C_{trace} + $C_{devices}$ = 20 + 46 = 66 pF (4.54)
Propagation delay due to speed of light = 10"/12" x 2 = 1.6 ns (4.55)
Delay due to capacitive load = (66 -25) /25 = 1.55 ns (4.56)
From Eq. (4.23) and Eq. (4.24),
Total propagation delay on a store = 1.54 + 1.67 ≈ 3 ns (4.57)

4.3.2.2 Load (RAM Provides Data)

Since the trace length is the same, C_{trace} = 20 pF (4.58)
Capacitive load due to devices,
$C_{devices}$ = C_{R3001} + C_{R3010} + C_{RAMin} + C_{374in} + C_{823}
$C_{devices}$ = 10 + 10 + 12 + 10 + 7 = 49 pF (4.59)
Total capacitance = C_{trace} + $C_{devices}$ = 20 + 49 = 69 pF
The RAM rated drive is 30 pF.
Extra load = Total capacitance - RAM rated drive = 69 - 30 = 39 pF (4.60)
Propagation delay due to capacitive load = 39/30 = 1.3 ns (4.61)
Propagation delay due to the path length = 1.6 ns (4.62)
Total propagation delay = 1.3 + 1.6 ≈ 3 ns (4.63)

9

4.3.3 Read and Write Control Deratings

For a through hole design, the effect of derating on the control signals will be more. This section calculates that effect. The trace length from the CPU to the RAMs is 9 inches for the layout shown in Figure 4.5. Each control signal is connected to 8 devices.

Number of RAM devices connected to each control signal = 12 (4.64)

Input capacitance of each RAM = 5 pF (4.65)

Total load capacitance = 12 x 5 = 60 pF (4.66)

The trace length = 9" (4.67)

Trace capacitance = 9" x 2 pF/inch = 18 pF (4.68)

Total load capacitance = 60 + 18 = 78 pF (4.69)

Extra load = 78 - 25 = 53 pF (4.70)

Derating due to capacitive load = 53 / 50 = 1.6 ns (4.71)

Propagation delay due to trace length = 9 / 12 x 2 ns/foot = 1.6 ns (4.72)

Total derating = 1.6 + 1.6 ≈ 3 ns (4.73)

4.4 Timing Equations for Cache Design

This section deals with the timing equations that enable us to determine the critical timing requirements of the static RAM that will be used as cache. These equations are based on the use of static RAMs (without built-in latches) as cache RAMs. The superscript 'd' in the following equations denote the deratings to be taken into account. The static RAM chosen for illustration here is a 16K x 4 IDT7198. **The board is assumed to be surface mount for all speeds of the R3001 except for the 16 MHz speed grade**. The deratings for the surface mount board is 2 ns and that for a through hole board (which is used for the 16 MHz R3001) is 3 ns. The deratings were derived from certain assumptions. The explanation and the methodology used is explained in the previous sections. In the following, a generalized equation is given followed by the timing requirements for different frequencies of the R3001. All calculations are based on the R3001 specifications for the four speed versions (16, 20, 25, and 33 MHz), which are found in the IDT data sheets.

Figures 4.6, 4.7, 4.8, and 4.9 show the timing diagrams of the R3001 when it is doing a data store followed by an instruction fetch. This is the worst case example and is chosen to determine the SRAM parameter requirements. Figure 4.6 shows the timing diagrams for an R3001 operating at 16 MHz. Figures 4.7,4.8, and 4.9 show the timing diagrams for an R3001 operating at 20 MHz, 25 MHz and 33 MHz respectively. The encircled numbers represent the equations presented in section 4.4. The timing diagram in conjunction with the equations are used to arrive at determining the timing requirements.

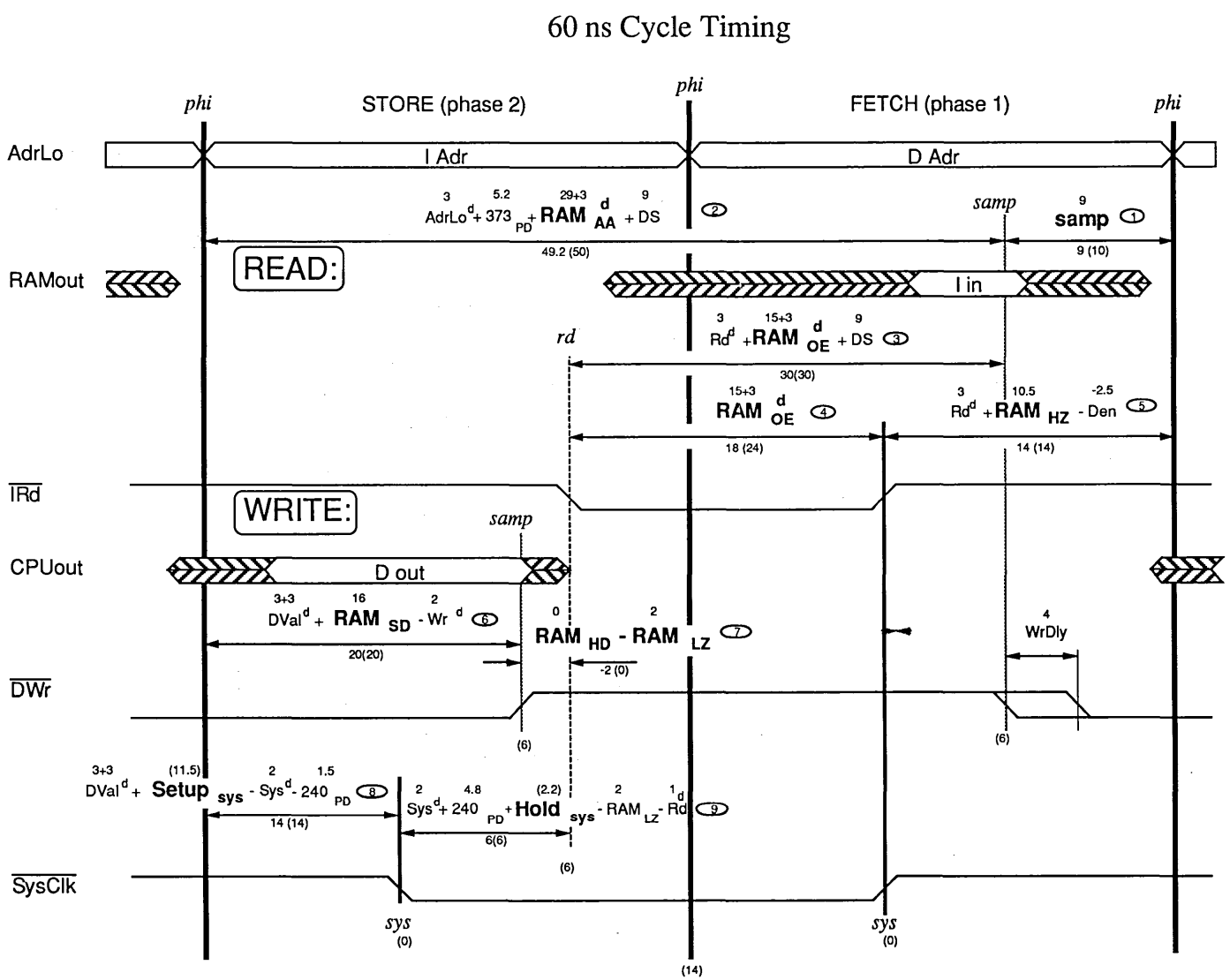


Figure 4.6. Cache Timing Diagram for a 16 MHz R3001

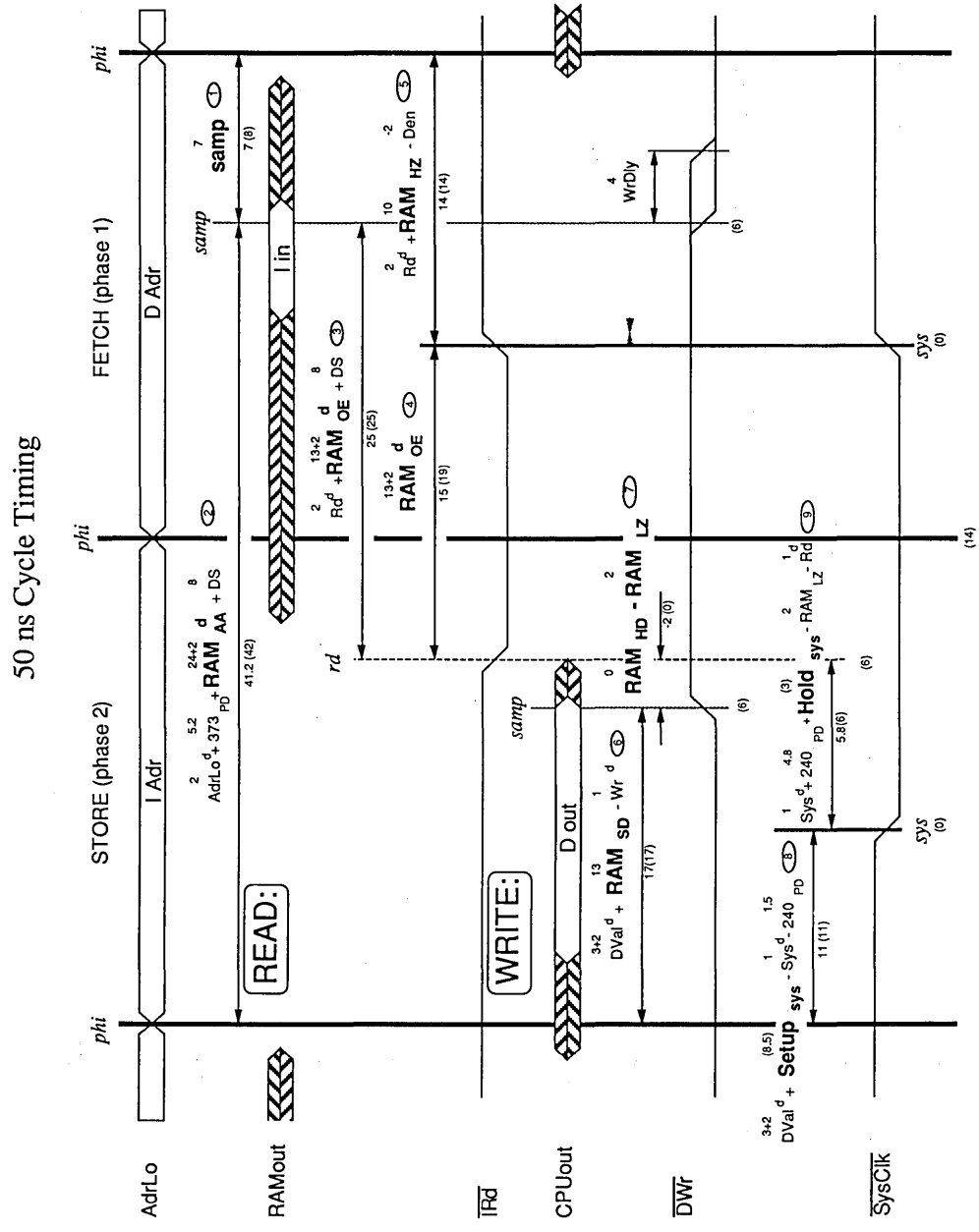


Figure 4.7. Cache Timing Diagram for a 20 MHz R3001

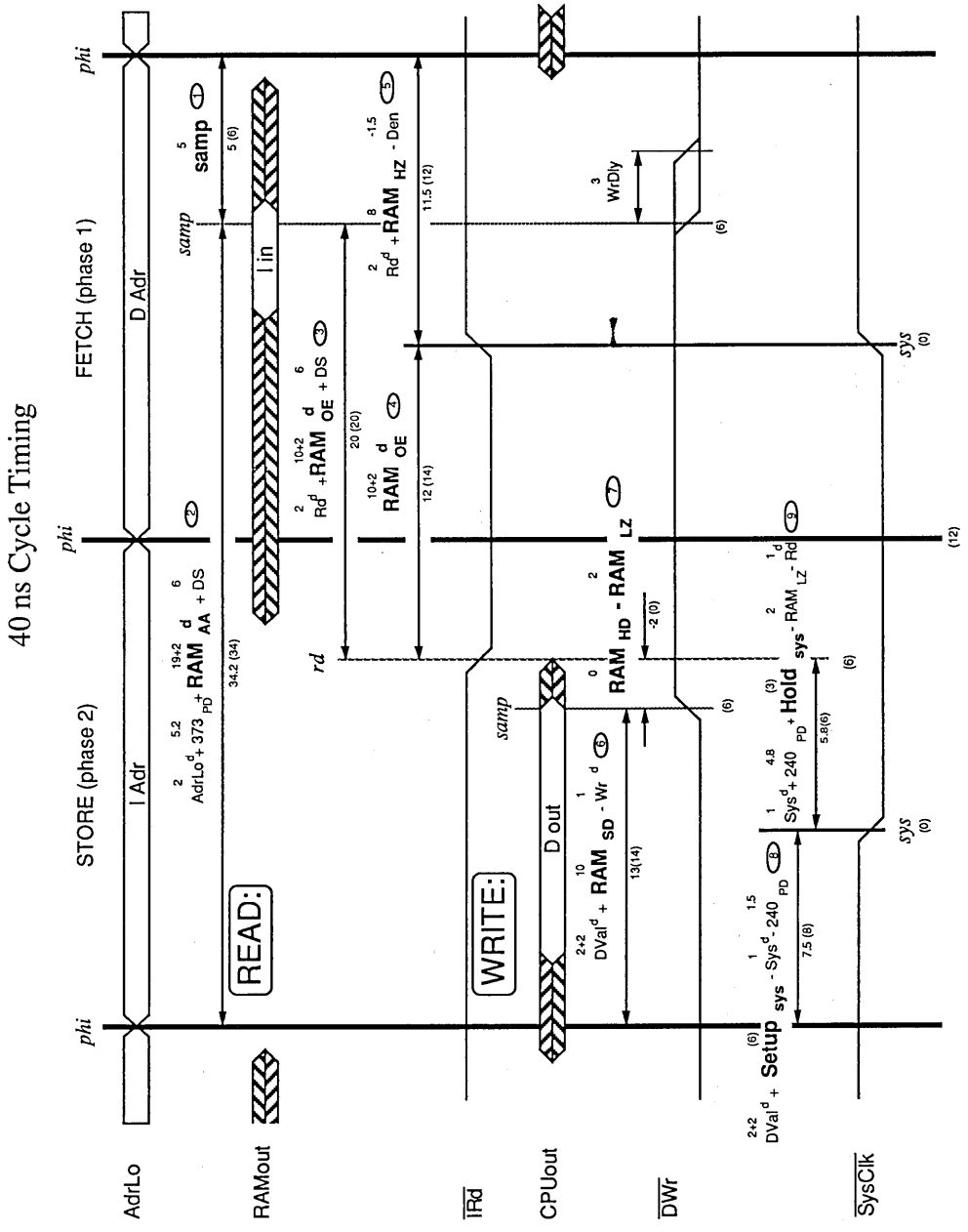


Figure 4.8. Cache Timing Diagram for a 25 MHz R3001

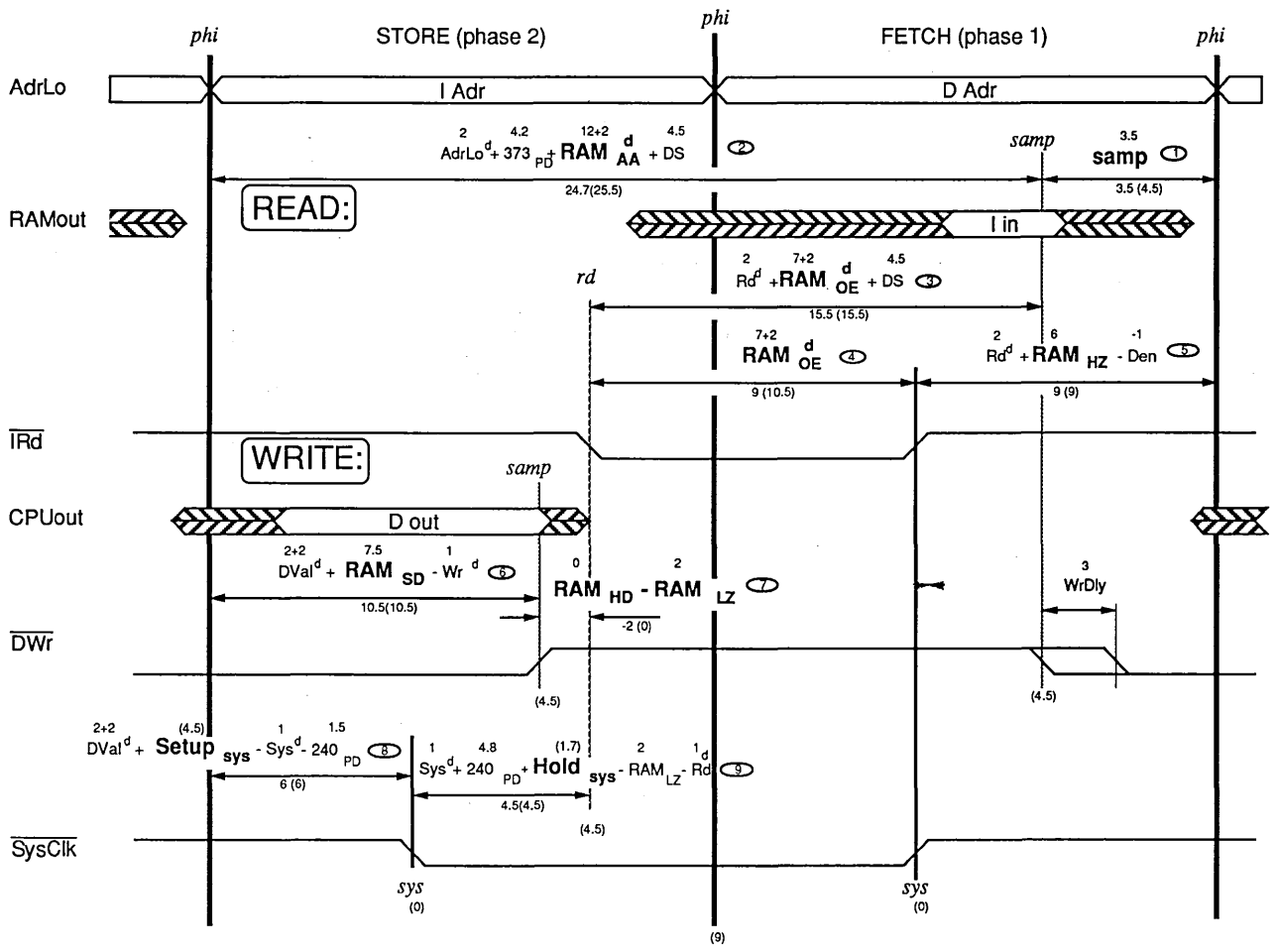


Figure 4.9. Cache Timing Diagram for a 33 MHz R3001

The following equations are used to determine the timing parameters for the static RAM so that they can function as cache for different operating frequencies of the R3001. The numbers at the left correspond to the encircled numbers in the timing diagrams. Equations 9 and 10 are not shown in the timing diagram but are included for completeness. The equations also use some R3001 parameters. These are listed in Table 4.2.

(1) Internal Sample to Phase Delay

This is the time that the processor needs to sample the incoming data. Typically, for the R3001, $t_{smp} \geq 5$.

(2) RAM Address Access Time

This equation is used to determine the Address Access time parameter requirements of the static RAM. From the timing diagram of Figure 4.9, it is easily calculated. As an example, let us calculate the address access time for a 33 MHz R3001. The total cycle time for a 33 MHz R3001 is 30 ns. If the processor's sample time requirement is met the time remaining in the cycle is 24 ns. This time the data has to be presented to the processor. The processor requires a data setup time of 4 ns. There is also a propagation delay through the latch for the address bus. For the 33 MHz part, a fast FCT373C is used which has a maximum propagation delay of 4.7 ns (See Table 4.3). The derating factors due to the capacitance and the trace length also have to be taken into account. Using all these factors, the equation is:

$t_{RAMAA} \leq t_{cyc} - t_{smp} - t_{DS} - t_{373PD} - t_{AdrLo}^{d} - t_{RAMAA}^{d}$

16 MHz R3001: $t_{RAMAA} \leq 60 - 10 - 9 - 5.2 - 3 - 3$
$t_{RAMAA} \leq 29.8$

20 MHz R3001: $t_{RAMAA} \leq 50 - 8 - 8 - 5.2 - 2 - 2$
$t_{RAMAA} \leq 24.8$

25 MHz R3001: $t_{RAMAA} \leq 40 - 6 - 6 - 5.2 - 2 - 2$
$t_{RAMAA} \leq 18.8$

33 MHz R3001: $t_{RAMAA} \leq 30 - 4.5 - 4.5 - 4.7 - 2 - 2$

$t_{RAMAA} \leq 12.3$

(3) Cache Enable to Sample

This equation is used to determine the system output enable(t_{OES}) requirements of the cache RAM. This should meet the processor's setup specification. The output enable time (t_{OE}) specifications for the RAM is tested for a voltage change of 200 mV (a fall from 1.732 V to 1.532 V for IDT RAMs). For a system, however, the voltage falls from approximately 3.3 V to 1.5 V. This fall time is usually a nanosecond. **Therefore the RAM specifications should take this system factor into consideration and specify the output enable time at least one nanosecond lower than the calculated timings.**

$t_{OES} \leq t_{cyc/2} - t_{RD}^{d} - t_{DS} - t_{sys\text{-}smp} + t_{sys\text{-}rd} - t_{OES}^{d}$

16 MHz R3001: $t_{OES} \leq 30 - 3 - 9 - 10 + 10 - 3$
$t_{OES} \leq 15$

20 MHz R3001: $t_{OES} \leq 25 - 2 - 8 - 8 + 8 - 2$
$t_{OES} \leq 13$

25 MHz R3001: $t_{OES} \leq 20 - 2 - 6 - 6 + 6 - 2$
$t_{OES} \leq 10$

33 MHz R3001: $t_{OES} \leq 15 - 2 - 4 - 4.5 + 4.5 - 2$
$t_{OES} \leq 7$

(4) Minimum Read Pulse Width

This timing requirement guarantees that the read pulse width generated by the processor is at least as long as the cache RAM output-enable time.

$t_{OES} \leq t_{cyc/2} - t_{sys\text{-}r}^{d} - t_{OES}^{d}$

16 MHz R3001: $t_{OES} \leq 30 - 10 - 3$
$t_{OES} \leq 17$

20 MHz R3001: $t_{OES} \leq 25 - 8 - 2$
$t_{OES} \leq 15$

25 MHz R3001: $t_{OES} \leq 20 - 6 - 2$
$t_{OES} \leq 12$

33 MHz R3001: $t_{OES} \leq 15 - 4.5 - 2$
$t_{OES} \leq 8.5$

(5) Read-Write I-Cache Data Bus Contention

This timing requirement ensures that the RAM output is tristated soon enough after the instruction read signal goes high. In the worst case, when the processor performs a store operation, no data contention occurs.

$t_{RAMHZ} \leq t_{sys} - t_{Rd}^{d} + DEn$

16 MHz R3001: $t_{RAMHZ} \leq 16 - 3 + (-2.5)$
$t_{RAMHZ} \leq 10.5$

20 MHz R3001: $t_{RAMHZ} \leq 14 - 2 + (-2)$
$t_{RAMHZ} \leq 10$

25 MHz R3001: $t_{RAMHZ} \leq 12 - 2 + (-1.5)$
$t_{RAMHZ} \leq 8.5$

33 MHz R3001: $t_{RAMHZ} \leq 9 - 2 + (-1)$
$t_{RAMHZ} \leq 6$

9

(6) Processor Data-Setup to End of Write

This enables the designer to determine whether the cache RAMs have adequate data setup time when the processor does a store operation. In the equation, the minimum derating is used on the write line i.e., tWr^d because that is the worst case assumption.

$tRAMDS \leq tcyc/2 - tsys\text{-}smp - tDVal - tDVal^d - tWr^d$

16 MHz R3001: $tRAMDS \leq 30 - 10 - 3 - 3 - (-2)$
$tRAMDS \leq 16$

20 MHz R3001: $tRAMDS \leq 25 - 8 - 3 - 2 - (-1)$
$tRAMDS \leq 13$

25 MHz R3001: $tRAMDS \leq 20 - 6 - 2 - 2 - (-1)$
$tRAMDS \leq 11$

33 MHz R3001: $tRAMDS \leq 15 - 4.5 - 2 - 2 - (-1)$
$tRAMDS \leq 7.5$

(7) Data Hold from End of Write

This parameter requirement guarantees that the data hold from end of write of the cache RAM is met when the processor or the read buffer is writing to the RAMs.

$tRAMHD \leq tRAMLZ$

16 MHz R3001: $tRAMHD \leq 2$

20 MHz R3001: $tRAMHD \leq 2$

25 MHz R3001: $tRAMHD \leq 2$

33 MHz R3001: $tRAMHD \leq 2$

(8) Data Setup to SysClk

This timing parameter ensures that the setup time into an external register (for the main memory interface) is sufficient enough for when the processor is doing a store. The data is clocked in the register on the rising edge of the buffered $\overline{SysOut}$ (through an inverting FCT240A). In this equation, $tsys(min)^d$ is used to insure worst case calculations.

$tsetupSys \leq tcyc/2 - tsys - tDVal - tDVal^d + tsys^d + t240PDmin$

16 MHz R3001: $tSetupSys \leq 30 - 16 - 3 - 3 + 2 + 1.5$
$tSetupSys \leq 11.5$

20 MHz R3001: $tSetupSys \leq 25 - 14 - 3 - 2 + 1 + 1.5$
$tSetupSys \leq 8.5$

25 MHz R3001: $tSetupSys \leq 20 - 12 - 2 - 2 + 1 + 1.5$
$tSetupSys \leq 6.5$

33 MHz R3001: $tSetupSys \leq 15 - 9 - 2 - 2 + 1 + 1.5$
$tSetupSys \leq 4.5$

(9) Data Hold from SysClk

This timing parameter is to guarantee that the hold time specification for an external register is met on a processor store. In this equation the minimum value of tRD^d is taken to insure worst case numbers.

$tHoldSys \leq tsys\text{-}rd^d - tsys^d - t240PDmax + tRAMLZ + tRd^d$

16 MHz R3001: $tHoldSys \leq 6 - 2 - 4.8 + 2 + 1$
$tHoldSys \leq 2.2$

20 MHz R3001: $tHoldSys \leq 6 - 1 - 4.8 + 2 + 1$
$tHoldSys \leq 3.2$

25 MHz R3001: $tHoldSys \leq 6 - 1 - 4.8 + 2 + 1$
$tHoldSys \leq 3.2$

33 MHz R3001: $tHoldSys \leq 4.5 - 1 - 4.8 + 2 + 1$
$tHoldSys \leq 1.9$

(10) Address Setup to End of Write

This equation enables us to determine the timing requirement for the RAM so that the address set up time is sufficient before the trailing edge of the write pulse.

$tRAMAW \leq tcyc - tsmp\text{-}sys - tAdrLo^d - t373PD + tWr^d$

16 MHz R3001: $tRAMAW \leq 60 - 10 - 3 - 5.2 + 3$
$tRAMAW \leq 44.8$

20 MHz R3001: $tRAMAW \leq 50 - 8 - 2 - 5.2 + 2$
$tRAMAW \leq 36.8$

25 MHz R3001: $tRAMAW \leq 40 - 6 - 2 - 5.2 + 2$
$tRAMAW \leq 28.8$

33 MHz R3001: $tRAMAW \leq 30 - 4.5 - 2 - 4.7 + 2$
$tRAMAW \leq 20.8$

(11) Write Hold Pulse-Width

This requirement guarantees that the cache RAMs minimum write pulse width specification is met.

$t_{RAMPW} \leq t_{cyc/2} - t_{WrDly}$

16 MHz R3001: $t_{RAMPW} \leq 30 - 5$
$t_{RAMPW} \leq 25$

20 MHz R3001: $t_{RAMPW} \leq 25 - 4$
$t_{RAMPW} \leq 21$

25 MHz R3001: $t_{RAMPW} \leq 20 - 3$
$t_{RAMPW} \leq 17$

33 MHz R3001: $t_{RAMPW} \leq 15 - 2$
$t_{RAMPW} \leq 13$

(12) Write Recovery Time

The write recovery time is the time between the write pulse going inactive and the change in address. This characteristic is usually specified by the SRAM manufacturer and is typically zero. This parameter is important in the R3001 cache interface and care must be taken to choose the proper part to prevent race conditions. In the R3001 cache design using the IDT7198 16K x 4 RAM, the latch enable is controlled by IClk/DClk and the write enable on the RAM is controlled by $\overline{IWr}$/$\overline{DWr}$. The timing diagram shows the relationship between the two clocks and the parameter TWR. Timing calculations below show that the write recovery specifications are not violated.

Derating calculations for DClk and $\overline{DWr}$

To calculate the effect of derating on the control signals DClk and $\overline{DWr}$, the following assumptions have been made.
1) The pin to pin variation on an R3001 device is 15 % for a 50 pF load. Under the maximum case, the deratings will vary from 1.7 to 2 ns for DClk and $\overline{DWr}$. Under the minimum case the deratings will vary from 0.58 to 0.625 ns.
2) The trace length for the $\overline{DWr}$ signal is 6 inches.
3) The trace length for the DClk signal is 2 inches.
4) The trace length of the address bus to the RAM is 4 inches.
5) Each IClk control signal is connected to four FCT373 devices.

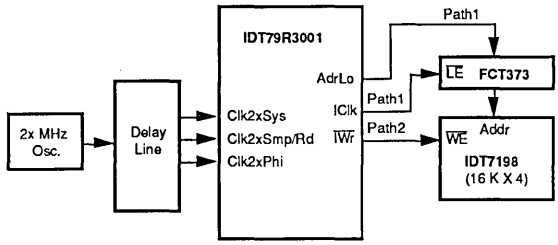


Figure 4.10a. Circuit Showing $\overline{IWr}$ and IClk Signals to Latch and SRAM

The input and ouput capacitances for the R3001, IDT7198, and FCT373 can be obtained from Table 4.1. Figure 4.10a is a simple circuit showing the connections of IClk and $\overline{IWr}$ from the R3001 to the latch enable (LE) on the FCT373 device and Write Enable ($\overline{WE}$) on the static RAM respectively. Figure 4.10b shows the tWR timings with respect to the data cache in an R3001 based system.

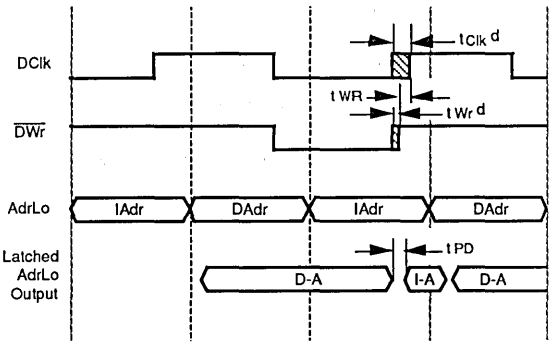


Figure 4.10b. Write Recovery Timing

To prove that the TWR parameter is not violated, the calculations are done as follows. The derating effects on the DClk and AdrLo signal should exceed that of the $\overline{DWr}$ signal. The calculations are similar to the derating calculations in previous sections. The minimum propagation delay through the latch is considered. The derating on the DClk signal coming out of the 79R3001 is lesser than that of the $\overline{DWr}$ signal. The reverse case is superfluous and in fact makes the situation better. The minimum and worst case derating effects on the same 79R3001 have been shown. This is because the write recovery time parameter must not be violated over the entire operating range.

9

Capacitive Derations:
(R3001 variations 15% 1.7ns - 2ns)

((Cdriver + Cload + Ctrace) - Crated)/25 * (Min Or Max) = CLD

Iclk ((10 + 4 * 10 + 2 * 2) - 25)/25 * **1.7** = 1.97ns

Iwr ((10 + 8 * 7 + 6 * 2) - 25)/25 * **2** = 4.24ns

RamAddr ((12 + 8 * 7 + 4 * 2) - 50)/25 * **0.5** = 0.52ns

Calculations :

Race path 1 : Iclkmin + Tpd(Ie) + RamAdrmin =
(1.97 + 2 + 0.52) = 4.49
Race path 2 : Iwrmax = 4.24

Path1 - Path2 > Twr

4.49 - 4.24 > 0

Capacitive Derations:
(R3001 variations 15% 0.58ns - 0.625ns)

((Cdriver + Cload + Ctrace) - Crated)/25 * (Min Or Max) = CLD

Iclk ((10 + 4 * 10 + 2 * 2) - 25)/25 * **0.58** = 0.58ns

Iwr ((10 + 8 * 7 + 6 * 2) - 25)/25 * **0.625** = 1.325ns

RamAddr ((12 + 8 * 7 + 4 * 2) - 50)/25 * **0.5** = 0.52ns

Calculations:

Race path 1 : Iclkmin + Tpd(Ie) + RamAdrmin =
(0.58 + 2 + 0.52) = 3.1ns
Race path 2 : Iwrmax = 1.325

Path1 - Path2 > Twr

3.1 - 1.325 = 1.775 > 0

From the above calculations and the RAM timing tables 4.3 and 4.4, it can be seen that the data setup to the processor is met. The output enable of the RAM which is controlled by $\overline{IRd}$ goes high and the RAM output starts to go tri-state. From the figure, the reader may correctly question whether the hold time requirements of the R3001 are met. It is indeed met by the capacitance on the bus and also due to the fact that CMOS devices are being used. The technical note entitled "Meeting Bus Hold for the R3001" gives a more detailed explanation.

Table 4.3 gives the timing data sheet for a typical SRAM device. The timing parameters correspond to a particular RAM configuration. Other RAM devices may have different timings for some of the parameters. However, there are certain timings that must be met. These critical parameters are listed in Table 4.4 and the unlisted parameters may vary a bit from device to device.

AC ELECTRICAL CHARACTERISTICS— COMMERCIAL TEMPERATURE RANGE

SYMBOL	PARAMETER	TEST CONDITIONS	16 MHz MIN.	16 MHz MAX.	20 MHz MIN.	20 MHz MAX.	25 MHz MIN.	25 MHz MAX.	UNIT
Clock									
TCkHigh	Input Clock High[2]	Transition < 5ns	12.5	—	10	—	8	—	ns
TCkLow	Input Clock Low[2]	Transition < 5ns	12.5	—	10	—	8	—	ns
TCkP	Input Clock Period[5]		30	500	25	500	20	500	ns
	Clk2xSys to Clk2xSmp/Rd[5]		0	Tcyc/4	0	Tcyc/4	0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi[5]		9	Tcyc/4	7	Tcyc/4	5	Tcyc/4	ns
Run Operation									
TDen	Data Enable[3]		—	–2	—	–2	—	–1.5	ns
TDDis	Data Disable[3]		—	–1	—	–1	—	–0.5	ns
TDVal	Data Vaild	Load = 25pF	—	3	—	3	—	2	ns
TWrDly	Write Delay	Load = 25pF	—	5	—	4	—	3	ns
TDS	Data Set-Up		9	—	8	—	6	—	ns
TDH	Data Hold		–2.5	—	–2.5	—	–2.5	—	ns
TCBS	CpBusy Set-Up		13	—	11	—	9	—	ns
TCBH	CpBusy Hold		–2.5	—	–2.5	—	–2.5	—	ns
TAcTy	Access Type [1:0]	Load = 25pF	—	7	—	6	—	5	ns
TAT2	Access Type [2]	Load = 25pF	17	—	14	—	12	—	ns
TMWr	Memory Write	Load = 25pF	1	27	1	23	1	18	ns
TExc	Exception	Load = 25pF	—	7	—	7	—	5	ns
Stall Operation									
TSAVal	Address Valid	Load = 25pF	—	30	—	23	—	20	ns
TSAcTy	Address Type	Load = 25pF	—	27	—	23	—	18	ns
TMRdI	Memory Read Initiate	Load = 25pF	1	27	1	23	1	18	ns
TMRd	Read Terminate	Load = 25pF	—	7	—	7	—	5	ns
TStl	Run Terminate	Load = 25pF	2	17	2	15	2	11	ns
TRun	Run Initiate	Load = 25pF	—	7	—	6	—	4	ns
TSMWr	Memory Write	Load = 25pF	1	27	1	23	1	18	ns
TSEc	Exception Valid	Load = 25pF	—	20	—	18	—	15	ns
TSEc	DMA Drive On	Load = 25pF	3	15	3	15	3	15	ns
TSEc	DMA Drive Off	Load = 25pF	—	10	—	10	—	10	ns
Reset Initialization									
TRST	Reset Pulse Width		6	—	6	—	6	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		140	—	140	—	140	—	µs
Capacitive Load Deration									
CLD	Load Derate		0.5	1	0.5	1	0.5	1	ns/25pF

NOTES:
1. All timings are referenced to 1.5V
2. The clock parameters apply to all three 2x clocks: Clk2xSys, Clk2xSmp/Rd and Clk2xPhi.
3. This parameteris guaranteed by design.
4. These parameters are illustrated in detail in the "IDT79R3001 Hardware Interface Guide".
5. Tcyc is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of Run, no two signals of a given device will derate by a difference greater than 15%.

Table 4.2. R3001 AC Specifications.*PLL: Phase Locked Loops

9

READ CYCLE TIMING SPECIFICATIONS

	16.7 MHz		20.0 MHz		25.0 MHz		33.0 MHz	
Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tRC	30	—	25	—	20	—	12	—
tAA	—	30	—	25	—	19	—	12[1]
tACS1	—	30	—	25	—	20	—	15
tCLZ1	5	—	5	—	5	—	2	—
tOES	—	15	—	13	—	10	—	7
tOLZ	5	—	5	—	5	—	3	—
tCHZ1	—	12	—	10	—	8	—	8
tOHZ	—	10	—	10	—	8	—	6
tOH	5	—	5	—	5	—	0	—
tPU	0	—	0	—	0	—	0	—
tPD	—	30	—	25	—	20	—	15

WRITE CYCLE TIMING SPECIFICATIONS

	16.7 MHz		20.0 MHz		25.0 MHz		33.0 MHz	
Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tWC	30	—	25	—	20	—	15	—
tCW1	25	—	21	—	17	—	15	—
tAW	25	—	21	—	17	—	15	—
tAS	0	—	0	—	0	—	0	—
tWP	25	—	21	—	17	—	13	—
tWR1	0	—	0	—	0	—	0	—
tWR2	0	—	0	—	0	—	0	—
tWHZ	—	18	—	16	—	8	—	6
tDW	16	—	13	—	11	—	7	—
tDH	0	—	0	—	0	—	0	—
tOW	5	—	5	—	5	—	5	—

Table 4.3. Static RAM Read and Write Timings to Work as Cache with the R3001

NOTE:
1. This assumes that an FCT373C with a TPD = 4.7 ns is used.

READ CYCLE TIMING SPECIFICATIONS

	16.7 MHz		20.0 MHz		25.0 MHz		33.0 MHz	
Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tRC	*	—	*	—	*	—	*	—
tAA	—	29.8	—	24.8	—	18.8	—	12.3
tACS1	—	*	—	*	—	*	—	*
tCLZ1	*	—	*	—	*	—	*	—
tOES	—	15	—	13	—	10	—	7
tOLZ	*	—	*	—	*	—	*	—
tCHZ1,2	—	*	—	*	—	*	—	*
tOHZ	—	10.5	—	10	—	8.5	—	6
tOH	*	—	*	—	*	—	*	—
tPU	*	—	*	—	*	—	*	—
tPD	—	*	—	*	—	*	—	*

WRITE CYCLE TIMING SPECIFICATIONS

	16.7 MHz		20.0 MHz		25.0 MHz		33.0 MHz	
Parameter	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
tWC	*	—	*	—	*	—	*	—
tCW1	*	—	*	—	*	—	*	—
tAW	44.8	—	36.8	—	28.8	—	19.8	—
tAS	*	—	*	—	*	—	*	—
tWP	25	—	21	—	17	—	13	—
tWR1	0	—	0	—	0	—	0	—
tWR2	0	—	0	—	0	—	0	—
tWHZ	--	*	--	*	--	*	--	*
tDW	16	—	13	—	11	—	7	—
tDH	*	—	*	—	*	—	*	—
tOW	*	—	*	—	*	—	*	—

Table 4.4. Static RAM Parameters to Work as Cache with the R3001

NOTE: All the parameters shown are the most allowable for maximum and minimum, respectively. Numbers not shown are not critical for the R3001 application.

Parameter	Load	Symbol	Min.	Max.
FCT373A Propagation Delay	50	t373 PD	—	5.2
FCT373A Latch Enable Delay	50	t373 LE	2	8.5
FCT373A Latch Enable Hold	50	t373 Hld	1.8	—
FCT240A Propagation Delay	50	t240 PD	1.5	4.8
FCT373C Propagation Delay	50	t373 PD	1.5	4.7
FCT240C Propagation Delay	50	t240 PD	1.5	3.7

Table 4.5. Timing Parameters of FCT Logic Devices

4.4.1 Legend

tRAMAA - RAM Access Time

tRAMOE - RAM Output Enable Time

tRAMHZ - RAM Output Low impedance to Output in High impedance

tRAMLZ - RAM Output in High impedance to output in Low impedance

tRAMHD - RAM Data Hold Time

tDS- R3001 Data Setup Time

tsys- Phase Difference between Clk2xSys and Clk2xPhi

trd- Phase Difference between Clk2xPhi and Clk2xSmp/Rd

tsmp- Phase Difference between Clk2xPhi and Clk2xSmp/Rd

tcyc- Cycle time of the R3001

t240PD - Propagation delay from Clk to Output of FCT240A

5.0 USING x16 LATCHED RAMS AS CACHE FOR THE R3001 ON A SURFACE MOUNT DESIGN

5.1 Assumptions for Surface Mount Design Layout Using x16 Latched RAMs as Cache for the R3001

In this chapter, the RAM timings are calculated for a 4K X 16 IDT71586 which have the latches built in. For the static RAMs with latches built in, the address access times tRAMAA, and the address setup to end of write tRAMAW will change from those of a regular static RAM. The propagation delay due to the latches is eliminated increasing the access time and the address setup to end of write by about 5 ns. In addition the board layout is different because the distances from the CPU to the RAM is reduced. This decreases the derating factors by a finite amount. This chapter calculates the derating factors for an R3001 cache design using the IDT71586 as cache. These are the following assumptions:

1) The trace has a capacitance of 2 pF/inch

2) The speed of light is 2 ns/foot in epoxy.

3) The R3001 speeds are specified with a loading of 25pF. For every additional 25pF, there is a delay of 1ns. Note that the cache control signals are specified with a 50pF load and derate 1ns/25pF after that.

4) The distances between the R3001 and the latches are approximately 5 inches.

5) The distances between the R3001 and the RAMs are approximately 2 inches each.

6) In all of the assumptions, it is assumed that a surface mount package is used. The input capacitance of the RAMs is a typical value (7 pF) for a PLCC package.

The instruction and data caches are designed using the IDT71586 latched SRAMs. The system memory is assumed to be 256 MB. Data parity on the R3001 is disabled. The upper four tag bits are also not compared and their comparision is turned off at reset time using pull down resistors on the Tag (31:28) bits. Therefore the cache data format is 48 bits wide. The instruction and data caches are built using six IDT71586 - three for instruction cache and three for the data cache.

5.2 Derating Calculations Using IDT71586 as Cache RAMs

The derating factors for the IDT71586 cache RAMs follow the same methodology as explained in Chapter 4. The cache size is 4K words for instruction and 4K words for data. The latches are eliminated. The derating factors for the address and data bus are calculated.

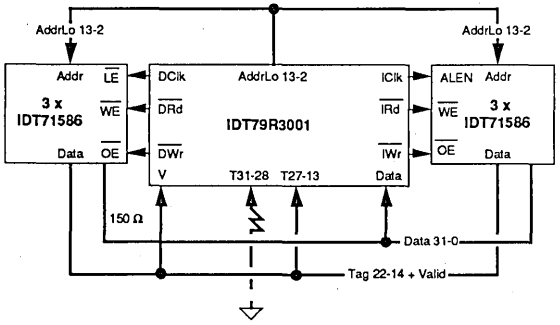


Figure 5.1. 71586 Used as Cache RAMs for the R3001 Cache Size = 16KB

Figure 5.1 shows a cache system for the R3001 with the latched RAMs i.e., IDT71586 as the cache. There are a total of 6 such devices required for 16KB each of instruction and data cache.

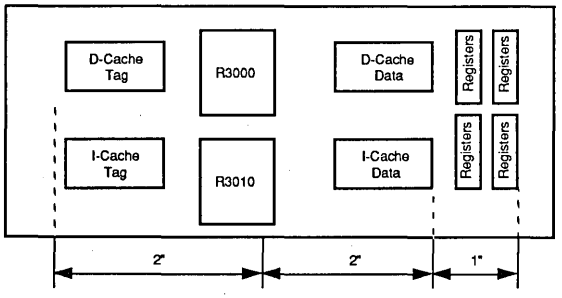


Figure 5.2. Surface Mount Board Layout of an R3001 System Using IDT71586 as Cache and Approximate Distances Between Devices

Figure 5.2 shows an example layout of an R3001 surface mount design board using latched RAMs (IDT71586) as cache for the R3001 system. The distance between the R3001 data pins and the caches is about 2 inches. The total trace length for the address bus and the data bus is about 4 inches each.

5.2.1 Address Bus Derating Calculations

Each AdrLo bus is connected to eight latched RAMs i.e., the IDT71586 and the address latch for main memory writes and reads. (Figure 5.3)

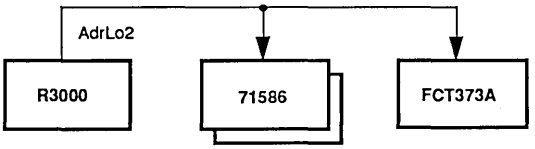


Figure 5.3. Number of Devices Address Bus is Connected To

Trace length from the CPU to the address latch (for main memory) = 4 inches (5.1)

Capacitance of the trace = C_{trace} = 4 x 2 pF/inch = 8 pF (5.2)

Input capacitance of the 373 latch = 10 pF (5.3)

Total input capacitance due to 6 RAM devices = 6 x 7 = 42 pF (5.4)

Total capacitance due to the load = 8 + 42 + 10 = 60 pF (5.5)

The rated R3001 load = 25 pF (5.6)

Extra loading on the R3001 = 60 - 25 = 35 pF (5.7)

The delay can be calculated as follows.

For every extra 25 pF of load, there is a delay of 1 ns (5.8)

From Eq. 5.7 and Eq. 5.8, delay due to the capacitive load = 35 / 25 = 1.4 ns (5.9)

The speed of light ≈ 2 ns/foot (5.10)

For a maximum path length of 3", delay = 3"/12" x 2 = 0.5 ns (5.11)

From Eq. 5.9 and Eq. 5.11,

Total propagation delay for the address bus = 1.4 + 0.5 ≈ 2 ns (5.12)

From the above calculations, it is seen that the derating on the address bus is 2 ns.

5.2.2 Data Bus Derating Calculations

From Figure 5.4, it is seen that the data bus is connected to the floating point unit (R3010), two 71586 devices, one read register (FCT374A), and one write register (FCT823B). As in the previous chapter where we considered a 16 K x 4 static RAM, we have to calculate the deratings for two cases: i) for an instruction fetch, and ii) for a data store.

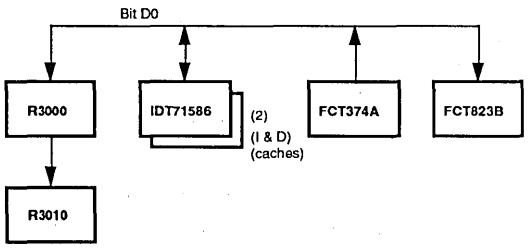


Figure 5.4. Devices Data Bus of the R3001 Is Connected to

5.2.2.1 Data Store (R3001 outputs data)

Each data bit is connected to two RAM devices — one for instruction and one for data.

The path length of the data bus = 4 inches. (5.13)

Trace capacitance of the data bus = 4 x 2 pF/inch = 8pF (5.14)

Capacitive loading on the data bus due to the different devices = 2 x C_{RAMin} + $C_{R3010in}$ + C_{374out} + C_{823in} = 2 x 7 + 10 + 12 + 10 = 46 pF (5.15)

Total capacitive load = $C_{devices}$ + C_{trace} = 46 + 8 = 54 pF (5.16)

Propagation delay due to speed of light = 4"/12" x 2 = 0.6 ns (5.17)

Delay due to capacitive load = (54 - 25) /25 = 1.16 ns (5.18)

Total delay = 1.16 + 0.7 = 1.8 ns ≈ 2 ns. (5.19)

5.2.2.2 Load Data Into R3001(RAM outputs data)

Since the trace length is the same, the trace capacitance C_{trace} = 8 pF. (5.20)

Capacitive load = $C_{R3001in}$ + $C_{R3010in}$ + $C_{71586in}$ + C_{374in} + C_{823out} (5.21)

$C_{devices}$ = 10 + 10 + 7 + 12 + 10 = 49 pF (5.22)

C_{total} = 49 + 8 = 57 pF (5.23)

The RAM rated drive = 30 pF (5.24)

Propagation delay due to extra capacitive loading = (57 - 30) / 25 = 1.08 ns (5.25)

Propagation delay due to path length = 0.8 ns (5.26)

Total propagation delay = 1.08 + 0.8 ≈ 2 ns (5.27)

5.2.3 Read and Write Control Derating Calculations

The effect of the capacitance on the control signals from the R3001 processor to the caches and the memory interface is considered here. The control signals on the R3001 are the $\overline{IRd}$, $\overline{DRd}$, $\overline{IWr}$, and $\overline{DWr}$ which control the instruction cache read, data cache read, instruction cache write, and data cache write respectively. The read and write control signals are connected to the output enable ($\overline{OE}$), and write enable ($\overline{WE}$) of the instruction and data cache respectively. Two control signals each are provided for the read and write operations of each of the caches. Assuming the use of a 4 K x 16 IDT71586 static RAM, each control signal is connected to 3 such static RAMs.

Number of devices (SRAM) connected to each control line = 3 (5.28)

Input capacitance of each device (SRAM) = 5 pF (5.29)

Total load capacitance = 3 x 5 = 15 pF (5.30)

Path length = 4" (5.31)

Trace Capacitance = 4 x 2 pF/in = 8 pF (5.32)

Total capacitance = 15 + 8 = 23 pF (5.33)

There is no extra capacitive loading here as the rated R3001 load is 50 pF.

Propagation delay due to the trace length = 0.8 ns (5.34)

Total propagation delay = 0.8 ≈ 1 ns (5.35)

5.3 Timing Equations for Cache Design

This section deals with the timing equations that enable us to determine the critical timing requirements of the static RAM that will be used as cache. These equations are based on the use of static RAMs without built-in latches as cache RAMs. The superscript 'd' in the following equations denote the deratings to be taken into account. The static RAM chosen for illustration here is a 4K x 16 IDT71586. The board is assumed to be surface mount for all speeds of the R3001. The deratings for the surface mount board is 2 ns. The deratings were derived from certain assumptions. The explanation and the methodology used is explained in the previous sections. In the following, a generalized equation is given followed by the timing requirements for different frequencies of the R3001. All calculations are based on the R3001 specifications for the four speed versions (16, 20, 25, and 33 MHz), which are found in the IDT data sheets.

(1) Internal Sample to Phase Delay

This is the time that the processor needs to sample the incoming data. Typically, for the R3001, $t_{smp} \geq 5$

(2) RAM Address Access Time

This equation is used to determine the Address Access time parameter requirements of the static RAM. From the timing diagram of Figure 5.4, it is easily calculated. The total cycle time for a 33 MHz R3001 is 30 ns. If the processor's sample time requirement is met, the time remaining in the cycle is 24 ns in which the data has to be presented to the processor. The processor requires a data setup time of 4 ns. The derating factors due to the capacitance and the trace length have also to be taken into account. Using all these factors, the equation is,

$t_{RAMAA} \leq t_{cyc} - t_{smp} - t_{DS} - t_{AdrLod} - t_{RAMAAd}$

16 MHz R3001: $t_{RAMAA} \leq 60 - 10 - 9 - 3 - 3$

$t_{RAMAA} \leq 35$

20 MHz R3001: $t_{RAMAA} \leq 50 - 8 - 8 - 2 - 2$

$t_{RAMAA} \leq 30$

25 MHz R3001: $t_{RAMAA} \leq 40 - 6 - 6 - 2 - 2$

$t_{RAMAA} \leq 24$

33 MHz R3001: $t_{RAMAA} \leq 30 - 4.5 - 4.5 - 2 - 2$

$t_{RAMAA} \leq 17$

(3) Cache Enable to Sample

This equation is used to determine the output enable requirements of the cache RAM and should meet the processor's setup specification. The output enable time for the latched RAM is specified by the manufacturer and tested for a voltage change of 200 mV (1.732 V to 1.532 V for IDT RAMs). For a system the voltage falls from a level of 3.3 V to 1.5V and the added fall time must be considered when specifying the RAM t_{OE} parameter. This fall time is approximately an additional nanosecond. Therefore the RAM t_{OE} parameter should be one nanosecond lower than the calculated numbers below.

$t_{OES} \leq t_{cyc/2} - t_{RD^d} - t_{DS} - t_{sys-smp} + t_{sys-rd} - t_{OES^d}$

16 MHz R3001: $t_{OES} \leq 30 - 2 - 9 - 10 + 10 - 3$

$t_{OES} \leq 16$

20 MHz R3001: $t_{OES} \leq 25 - 1 - 8 - 8 + 8 - 2$

$t_{OES} \leq 14$

25 MHz R3001: $t_{OES} \leq 20 - 1 - 6 - 6 + 6 - 2$

$t_{OES} \leq 11$

33 MHz R3001: $t_{OES} \leq 15 - 1 - 4 - 4.5 + 4.5 - 2$

$t_{OES} \leq 8$

(4) Minimum Read Pulse Width

This timing requirement guarantees that the read pulse width generated by the processor is at least as long as the cache RAM output enable time.

$t_{OES} \leq t_{cyc/2} - t_{sys-rd} - t_{OES^d}$

16 MHz R3001: $t_{OES} \leq 30 - 10 - 3$

$t_{OES} \leq 17$

20 MHz R3001: $t_{OES} \leq 25 - 8 - 2$

$t_{OES} \leq 15$

25 MHz R3001: $t_{OES} \leq 20 - 6 - 2$

$t_{OES} \leq 12$

33 MHz R3001: $t_{OES} \leq 15 - 4.5 - 2$

$t_{OES} \leq 8.5$

(5) Read-Write I-Cache Data Bus Contention

This timing requirement ensures that the RAM output is tri-stated soon enough after the instruction read signal goes high. In the worst case, when the processor performs a store operation, no data contention occurs.

$t_{RAMHZ} \leq t_{sys} - t_{Rd^d} + t_{DEn}$

16 MHz R3001: $t_{RAMHZ} \leq 16 - 2 + (-2.5)$

$t_{RAMHZ} \leq 11.5$

20 MHz R3001: $t_{RAMHZ} \leq 14 - 1 + (-2)$

$t_{RAMHZ} \leq 11$

25 MHz R3001: $t_{RAMHZ} \leq 12 - 1 + (-1.5)$

$t_{RAMHZ} \leq 9.5$

33 MHz R3001: $t_{RAMHZ} \leq 9 - 1 + (-1)$

$t_{RAMHZ} \leq 7$

(6) Processor Data-Setup to End of Write

This enables the designer to determine whether the cache RAMs have adequate data setup time when the processor does a store operation. In the equation, the

9

minimum derating is used on the write line i.e., tWr^d because that is the worst case assumption.

$tRAMDS \leq tcyc/2 - tsys\text{-}smp - tDVal - tDVal^d - tWr^d$

16 MHz R3001: $tRAMDS \leq 30 - 10 - 3 - 3 - (-2)$

$tRAMDS \leq 16$

20 MHz R3001: $tRAMDS \leq 25 - 8 - 3 - 2 - (-1)$

$tRAMDS \leq 13$

25 MHz R3001: $tRAMDS \leq 20 - 6 - 2 - 2 - (-1)$

$tRAMDS \leq 11$

33 MHz R3001: $tRAMDS \leq 15 - 4.5 - 2 - 2 - (-1)$

$tRAMDS \leq 7.5$

(7) Data Hold from End of Write

This parameter requirement guarantees that the data hold from end of write of the cache RAM is met when the processor or the read buffer is writing to the RAMs.

$tRAMHD \leq tRAMLZ$

16 MHz R3001: $tRAMHD \leq 2$

20 MHz R3001: $tRAMHD \leq 2$

25 MHz R3001: $tRAMHD \leq 2$

33 MHz R3001: $tRAMHD \leq 2$

(8) Data Setup to SysClk

This timing parameter ensures that the setup time into an external register (for the main memory interface) is sufficient enough for the case when the processor is doing a store. The data is clocked in the register on the rising edge of the buffered $\overline{SysOut}$ (through an inverting FCT240A). In this equation, $tsys(min)^d$ is used to insure worst case calculations.

$tsetupSys \leq tcyc/2 - tsys - tDVal - tDVal^d + tsys^d + t240PDmin$

16 MHz R3001: $tSetupSys \leq 30 - 16 - 3 - 3 + 2 + 1.5$

$tSetupSys \leq 11.5$

20 MHz R3001: $tSetupSys \leq 25 - 12 - 3 - 2 + 1 + 1.5$

$tSetupSys \leq 10.5$

25 MHz R3001: $tSetupSys \leq 20 - 12 - 2 - 2 + 1 + 1.5$

$tSetupSys \leq 6.5$

33 MHz R3001: $tSetupSys \leq 15 - 9 - 2 - 2 + 1 + 1.5$

$tSetupSys \leq 4.5$

(9) Data Hold from SysClk

This timing parameter is to guarantee that the hold time specification for an external register is met on a processor store. In this equation the minimum value of tRD^d is taken to insure worst case numbers.

$tHoldSys \leq tsys\text{-}rd - tsys^d - t240PDmax + tRAMLZ + tRd^d$

16 MHz R3001: $tHoldSys \leq 6 - 2 - 4.8 + 2 + 1$

$tHoldSys \leq 2.2$

20 MHz R3001: $tHoldSys \leq 6 - 1 - 4.8 + 2 + 1$

$tHoldSys \leq 3.2$

25 MHz R3001: $tHoldSys \leq 6 - 1 - 4.8 + 2 + 1$

$tHoldSys \leq 3.2$

33 MHz R3001: $tHoldSys \leq 4.5 - 1 - 4.8 + 2 + 1$

$tHoldSys \leq 1.9$

(10) Address Setup to End of Write:

This equation enables us to determine the timing requirement for the RAM so that the address set up time is sufficient before the trailing edge of the write pulse.

$tRAMAW \leq tcyc - tsmp\text{-}sys - tAdrLo^d + tWr^d$

16 MHz R3001: $tRAMAW \leq 60 - 10 - 3 + 2$

$tRAMAW \leq 49$

20 MHz R3001: $tRAMAW \leq 50 - 8 - 2 + 1$

$tRAMAW \leq 41$

25 MHz R3001: $tRAMAW \leq 40 - 6 - 2 + 1$

$tRAMAW \leq 33$

33 MHz R3001: $tRAMAW \leq 30 - 4.5 - 2 + 1$

$tRAMAW \leq 24.5$

(11) Write Hold Pulse Width:

This requirement guarantees that the cache RAMs minimum write pulse width specification is met.

$t_{RAMPW} \leq t_{cyc/2} - t_{WrDly}$

16 MHz R3001: $t_{RAMPW} \leq 30 - 5$

$t_{RAMPW} \leq 25$

20 MHz R3001: $t_{RAMPW} \leq 25 - 4$

$t_{RAMPW} \leq 21$

25 MHz R3001: $t_{RAMPW} \leq 20 - 3$

$t_{RAMPW} \leq 17$

33 MHz R3001: $t_{RAMPW} \leq 15 - 2$

$t_{RAMPW} \leq 13$

From the above calculations and Figure 5.3, it can be seen that the data setup to the processor is met. The output enable of the RAM which is controlled by $\overline{IRd}$ goes high and the RAM output starts to go tri-state. From the figure, the reader may correctly question whether the hold time requirements of the R3001 are met. It is indeed met by the capacitance on the bus and also due to the fact that CMOS devices are being used. The technical note entitled "Meeting Bus Hold for the R3001" gives a more detailed explanation.

From Figure 5.5, it is clearly seen that the address setup and hold time for the latched RAMs are met by using IClk to capture the instruction address. Figure 5.5 is to illustrate the timings for a 25 MHz IDT71586 latched RAM. Similar timing diagrams can be drawn to verify the setup and hold times for R3001 operating at different frequencies.

5.3.1 Legend

t_{RAMAA} - RAM Access Time

t_{RAMOE} - RAM Output Enable Time

t_{RAMHZ} - RAM OutPut Low impedance to Output in High impedance

t_{RAMLZ} - RAM Output in High impedance to output in Low impedance

t_{RAMHD} - RAM Data Hold Time

t_{DS} - R3001 Data Setup Time

t_{sys} - Phase Difference between Clk2xSys and Clk2xPhi

t_{rd} - Phase Difference between Clk2xPhi and Clk2xRd

t_{smp}- Phase Difference between Clk2xPhi and Clk2xSmp

t_{cyc} - Cycle time of the R3001

$t_{smp-rd} = t_{smp} - t_{rd}$

t_{240PD} - Propagation delay from Clk to Output of FCT240A

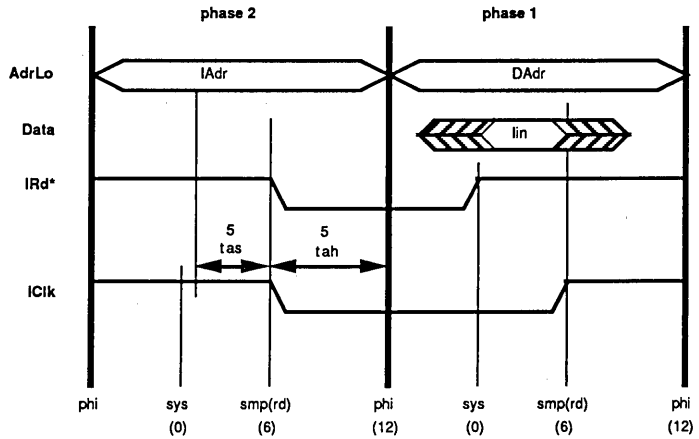


Figure 5.5. Address Setup and Hold Timing for a Latched RAM (25 MHz R3001)

Parameter	Description	16 MHz		20 MHz		25 MHz		33 MHz	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
t_{RC}	Read cycle	35	--	30	--	25		15	
t_{CH}	ALEN high	10		10		10		8	
t_{CL}	ALEN low	10		10		10		8	
t_{AS}	Adr Latch set-up	5		5		5		4	
t_{AH}	Adr Latch Hold	5		5		5		4	
t_{AA}	Address Access	--	35	--	30	--	24	--	17
t_{ACE}	Chip Enable access	--	35	--	30	--	25	--	17
t_{OES}	Output enable	--	16	--	14	--	11	--	8
t_{CLZ}	CE to out in LZ	3	--	3	--	3	--	3	--
t_{OLZ}	OE to out in LZ	2	--	2	--	2	--	2	--
t_{CHZ}	CE to out in HZ	--	25	--	22	--	20	--	15
t_{OHZ}	OE to out in HZ	--	11	--	11	--	9	--	7
t_{OH}	Output Hold from address in change	3	--	3	--	3	--	3	--

Table 5.1. Read Cycle Timings for an IDT Static RAM with Latches

Parameter	Description	16 MHz		20 MHz		25 MHz		33 MHz	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
t_{WC}	Write cycle	35	--	30	--	25		15	
t_{CH}	ALEN high	10		10		10		8	
t_{CL}	ALEN low	10		10		10		8	
t_{AS}	Adr Latch set-up	5		5		5		4	
t_{AH}	Adr Latch Hold	5		5		5		4	
t_{AW}	Address to end of write	35	--	30	--	25		15	
t_{ASW}	Address set-up	0	--	0	--	0	--	0	--
t_{WP}	Write pulse width	25	--	20	--	17	--	11	--
t_{CW}	CE to end of write	25	--	20	--	20	--	11	--
t_{WR}	Write recovery	0	--	0	--	0	--	0	--
t_{WHZ}	Write to out in HZ	--	15	--	15	--	13	--	8
t_{DW}	Data setup	16	--	13	--	11	--	7	--
t_{DH}	Data Hold	0	--	0	--	0	--	0	--
t_{OW}	Out active from end of write	5	--	5	--	5	--	5	--

Table 5.2. Write Cycle Timings for an IDT Static RAM with Latches

REFERENCES

1) Kane, Gerry., "mips RISC ARCHITECTURE,"
Prentice Hall Inc., N.J, 1988

2) IDT RISC R3001 Microprocessor Interface Guide,
April 1990

3) IDT RISC R3001 Family Data Sheets, *1990*

4) IDT Data Book, *1989*

5) IDT Data Book Supplement, *1989*

Integrated Device Technology, Inc.

A POWERFUL DEVELOPMENT TOOL FOR THE IDT 79R3000 RISC FAMILY

CONFERENCE PAPER CP–01

AS PRESENTED AT SOUTHCON '89

By Philip Bourekas

INTRODUCTION

The inherent flexibility of the IDT 79R3000 family architecture allows system designers to make a wide variety of system tradeoffs. Therefore, it is extremely important that effective development tools be available for each step of the design. This need starts when the initial architectural decisions and performance simulations are made and carries through to the final stages of hardware–software integration and system diagnostics development.

This paper describes the System Programmer's Package (SPP) for the IDT 79R3000, an effective set of tools available for each part of the design process. These tools allow designers to make tradeoffs in both hardware and software which result in systems with very cost–effective performance, and also speeds the overall development of the target application.

PROCESSOR OVERVIEW

The architecture of the IDT 79R3000 family is partitioned into a processor and a separate high–performance floating point unit. The processor, which can sustain 20 VAX MIPS performance, consists of two tightly–coupled processors implemented on a single ship. The first processor is a full 32–bit CPU utilizing RISC techniques to achieve a new standard of microprocessor performance; the second processor is a system control coprocessor (CP0), containing a Translation Lookaside Buffer (TLB) and control registers to support a 4 GByte virtual memory subsystem. Also integrated onto the processor chip is a dual–cache controller which controls separate direct–mapped instruction and data caches. Each cache can independently vary in size from 4K bytes through 256K bytes, with independently selectable block refill sizes of 1 to 32 words. The processor achieves 200 MBytes/ second cache bandwidth at 25 MHz. Figure 1 illustrates the interaction between the CPU, its coprocessors, and the memory subsystems.

The full performance of the 79R3000 is achieved by the proper integration of software and hardware. The 79R3000 contains a highly efficient five–stage pipeline, with each stage of the pipeline controlling a different CPU resource. Optimizing compiler technology serves to both reduce the number of instructions required to perform a given task, and also serves to efficiently schedule instructions. This eliminates processor stalls which might arise if a hardware interlock is activated and eliminates NOP instructions during the latency cycles of LOAD or BRANCH operations.

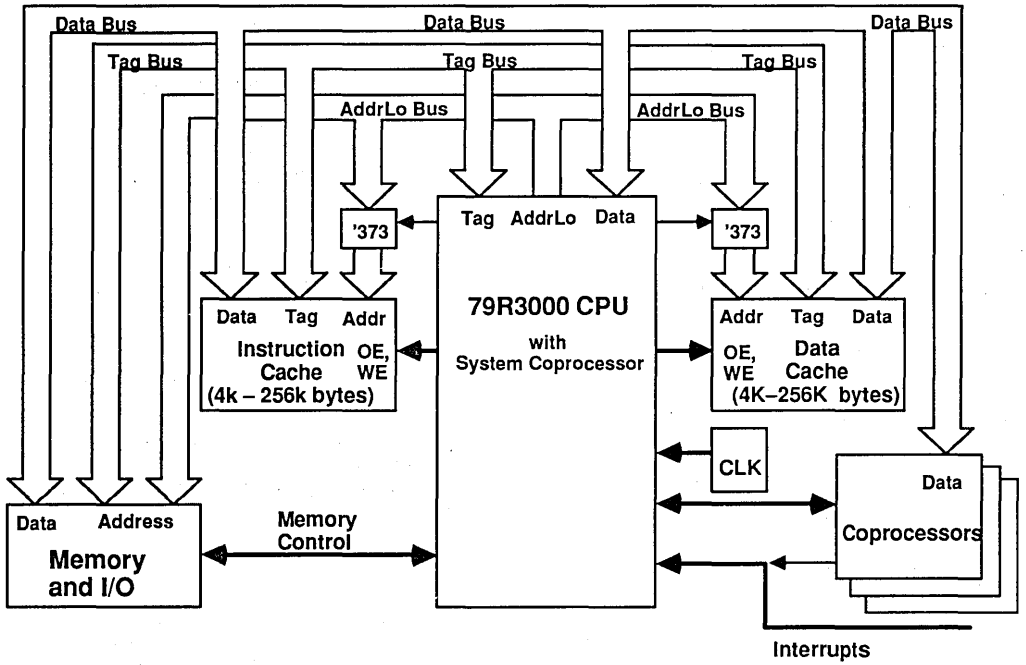


Figure 1. 79R3000 Based System

9

ARCHITECTURAL DECISION PHASE

In order to determine the system configuration which yields the best price/performance tradeoff for a given application, the system designer has to make a number of decisions. For example, the designer needs to select the cache configuration which best supports the application requirements. The 79R3000 allows the designer to choose various cache depths from 4K bytes to 256K bytes, and to select an appropriate block refill size, for both the instruction and data caches.

The optimal cache size and block size is a function of the locality of the typical programs running on the system, as well as the CPU sub-system's latency to main memory when a cache miss occurs. In general, the longer the latency to main memory the longer the CPU will be stalled for a cache miss and thus the more severe the penalty. This means that in order to overcome the effects of a long memory latency, a large cache and a fairly large block size are needed to support high performance. Systems with a relatively low latency to main memory may be able to tolerate a smaller, less expensive cache subsystem and a smaller block size.

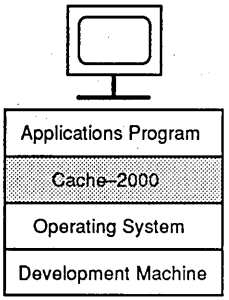


Figure 2. Using Cache-2000 to Determine Optimum System Configuration

Before going to the expense of drawing schematics or building boards, the system designer can use a set of programs contained in the SPP which simulates various cache sizes and refill sizes. Since the typical bus latency of the target system is included during the simulations, a relatively accurate model of anticipated performance is available. This capability is contained in the Cache-2000 module as part of the SPP; provided in source form, this package allows the system designer to describe the overall system and then benchmark typical target software (as shown in Figure 2). In this manner, a decision about the allowable tradeoffs in cost and performance can be made before the detailed design is begun.

SOFTWARE DEVELOPMENT PHASE

Software development is increasingly a bottleneck in system development. The SPP contains a number of utilities which facilitate the parallel development of software with hardware, allowing much of the software can be developed long before the target hardware is available.

Key to this is the stand-alone software simulator package, Sable, which models the CPU, FPA, and the entire memory hierarchy of TLB, Cache, main memory, disk and system console. Sable simulates the execution of 79R3000 instructions in an environment which mimics the target system. Further customization is possible, as the simulation package is provided in source form.

While Sable models things such as disks simply as a source or sink for data files without modeling the physical performance of a disk (latency, etc.), Sable does provide sophisticated models of the TLB and cache. Sable keeps track of the contents of the TLB and of each cache entry after every cycle. This allows debugging of kernels, diagnostics and other programs which might require knowledge of the privileged, supervisor state as well as facilitates debugging of user programs which may execute on top of an operating system.

Thus, by using the Sable architecture simulator, the software for the target system can be developed concurrently with the hardware itself. It is worth noting that MIPS Computer Systems uses Sable to debug its operating systems for new computers, and typically has the software working months in advance of the target hardware.

Other utilities are provided to minimize the software development effort. The Stand-Alone I/O library, SAIO, is designed to perform the functions of the stdio standard "C" library. The SAIO library contains Unix system-like routines that access disks, Ethernet, tapes, and UART devices and also includes routines to perform fault handling and cache flushing. The stand-alone compiler system uses the SAIO library to produce code that will execute in the target stand-alone environment. The entire stand-alone applications program is then constructed "on-top" of the modules provided by the SPP, as shown in Figure 3.

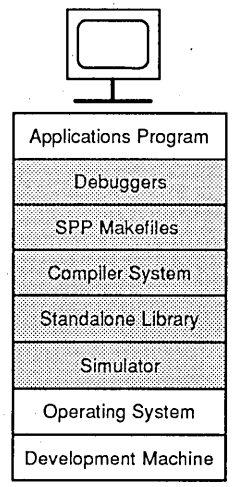


Figure 3. Using SPP to Develop Stand-Alone Applications on the Host System

Code can be developed in assembly language, "C", Pascal, Fortran, or even PL/1 or Cobol; the software development tools work equally well with any of MIPS standard languages. MIPS simulation debugger, sdbx, provides source level debugging across the entire language suite, tracing every machine action back to the original source statement. Sdbx provides the ability to examine and modify the state of the machine as well as standard debug facilities such as single-step or breakpoint.

Finally, another utility that comes standard with the MIPS RISC/os UNIX operating system, called PIXIE, provides extensive profiling of the target code. PIXIE dynamically profiles the execution of

the target software, so that the performance of the application can be tuned and so that all of the modules can be fully tested. PIXIE works on the binary image of the program, and thus does not require modification to the source code in order to be used.

HARDWARE/SOFTWARE INTEGRATION

The SPP toolkit also benefits the process of integrating software onto the target hardware. The SPP provide utilities which connect the target to the host development machine for final debugging, and also provides a number of software modules designed to be incorporated into the target to facilitate integration.

The SPP includes a PROM monitor which provides system diagnostics and initialization of the machine. Provided in source form, the monitor is designed to initialize the cache, reset the machine state, and transfer control either to a shell program or the applications program. The monitor is responsible for storing communications parameters for the console port and maintaining other system information. Utilities are provided to facilitate the compilation of the monitor program and to help burn it into PROMs. The PROM monitor allows basic commands to examine and modify system memory, and provides primitive debug support which is typically retained in the customer end product.

In addition to monitor functions, the PROM monitor also contains a set of power–on diagnostics which exercises the basic functionality of the CPU, FPA and memory. These diagnostics are also often incorporated into the customer end product. Once the target system can correctly execute the diagnostic suite, the application is known to communicate effectively with the cache and the basic operation of the CPU subsystem is confirmed.

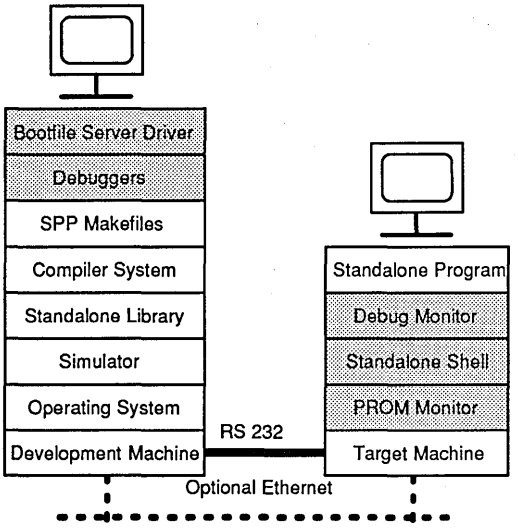


Figure 4. Using the SPP During Hardware/Software Integration

The Monitor and Stand–Alone Shell, SASH, extends the functionality of the simple PROM monitor. The monitor functions are partitioned between the PROM Monitor and SASH so that basic functions provided by the PROM monitor can be kept in a small PROM, while the extended SASH functions can then be executed out of RAM.

The 'dbgmon' debug monitor is designed to be incorporated into the target machine. It provides hooks which allow the host to remotely debug the target machine, and works with the Mips host system debugger, pdbx, which is a source level symbolic debugger configured to operate remotely. This duo performs remote debugging, including the ability to breakpoint, single step, and examine the execution state of the target. Beyond disassembly, the debugger maps machine instructions to the source statements which generated them, and prints variables and addresses symbolically.

Finally, in order to debug the target through the host development vehicle, communications software is also provided. Two RS232 ports in the target system aid development; one for the target systems' console monitor and one for the remote debug access to the target from the host. In addition to RS232 capability, drivers are also provided for Ethernet. Once the initial debug of the program is completed, Ethernet can be used to greatly speed the transfer of programs and data from the host. A Bootfile Server Driver is provided on the host to download bootable images to the target across the Ethernet. The SPP provides driver routines in source form, so that if necessary the drivers can be modified. Figure 4 illustrates the use of the SPP in debugging the stand–alone application on the target system.

HOST DEVELOPMENT SYSTEMS

The SPP has been designed to run on a variety of development hosts, provided that the host utilizes the R3000 processor. By running on a native mode platform, using a toolkit that was developed by the same people that developed the processor itself, there is no danger of errors in the interpretation of the workings of the processor.

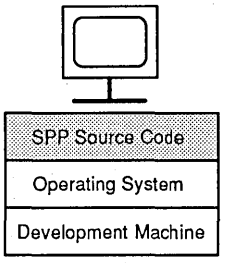


Figure 6. The Systems Programmer's Package for the IDT 79R3000

9

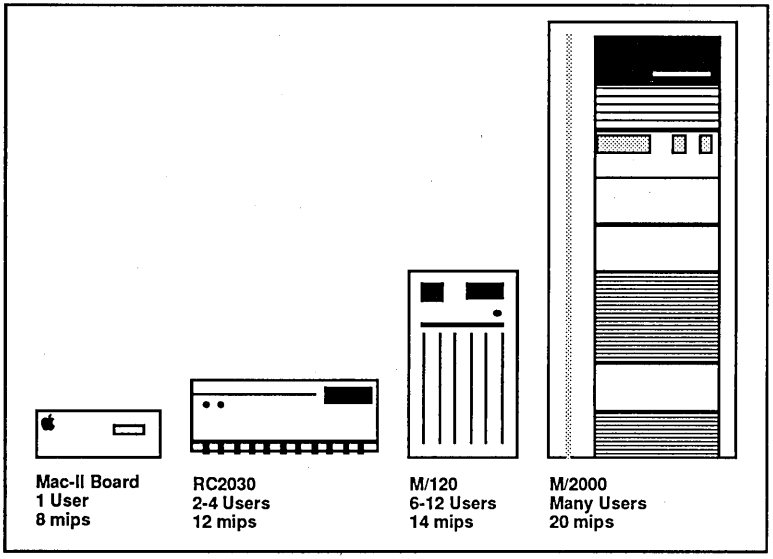


Figure 5. Range of SPP Host Development Systems for the IDT 79R3000

Platforms for the SPP include MIPS' M Series of RISC computers, which are powerful (10 to 20 VAX MIPS) multi-user UNIX environments, and also includes the 7RS201 Macintosh-II development card available from IDT. The choice of host development machine is primarily determined by the number of developers of the application. Figure 5 illustrates the range of SPP hosts available.

CONCLUSIONS

The IDT 79R3000 is a very powerful microprocessor based upon RISC principles. The processor architecture is supported by a very powerful toolset, the SPP, which supports product development during all phases of the design process.

During the initial architectural phase, the SPP provides tools which enable the system designer to make appropriate tradeoffs in cache design to achieve very cost effective performance for the end application.

Software development can then proceed concurrently with hardware development. The SPP includes language and library tools to facilitate development, as well as a powerful symbolic debugger for the software debug effort. An entire simulation environment simulates the processor, memory, and I/O subsystems. It is used to fully simulate the target system before it is even fully designed, and allows the target operating system and software to be debugged on an existing, functional platform.

Finally, the process of integrating software onto the target hardware is facilitated by modules included in the SPP. Source code for communications drivers, diagnostics, and a debugger and system monitor allow very powerful debug support as the software is brought up on the target hardware.

Altogether, the SPP enables applications to be developed for the 79R3000 in a very timely fashion, and allows the system designers to concentrate on developing the target system rather than on developing tools. The SPP provides the user with total control over the development process. Its power has been demonstrated at MIPS Computer Systems, where it is used to develop operating systems for new computers.

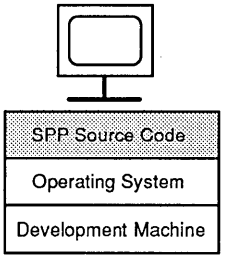


Figure 6. The Systems Programmer's Package for the IDT 79R3000

UNIX is a trademark of AT&T
VAX is a trademark of Digital Equipment Corporation
Macintosh is a trademark of Apple Computer
MIPS and RISC/os are trademarks of MIPS Computer Systems

Integrated Device Technology, Inc.

DEVELOPING APPLICATIONS FOR THE IDT 79R3000 RISC MICROPROCESSOR

AS PRESENTED AT NORTHCON '89

CONFERENCE PAPER CP–02

By Philip Bourekas

INTRODUCTION

Today's high–speed RISC microprocessors bring new levels of performance to todays' applications. This performance is a blend of the raw system speed and of the architecture of the processor. However, these processors require a different design methodology than the familiar techniques applicable to lower speed CISC–type microprocessors. Todays' processors feature tools rich in software content, appropriate to the software–intensive nature of RISC.

The IDT 79R3000 RISC Microprocessor family is richly supported by a wide variety of development tools, benefiting all stages of the development process. These tools allow designers to maximize performance, minimize cost, and significantly reduce time to market. Additionally, the tools are flexible enough to be well suited to the requirements of either reprogrammable or embedded type applications.

ARCHITECTURE OVERVIEW

The IDT 79R3000 is a high–performance microprocessor using RISC techniques to achieve 27 MIPS performance in a 33MHz system. The 79R3000 architecture was developed by MIPS Computer Systems as an evolution of work originally begun at Stanford University, and is a second generation implementation of the MIPS instruction set architecture.

The 79R3000 achieves this high–performance through the combination of low average cycles per instruction (typical of RISC processors) and high instruction and data bandwidth. The 79R3000 consists of two tightly–coupled processors on a single chip. The 32–bit integer RISC CPU is complemented by CP0, the on–chip system control coprocessor containing an MMU with 64–entry, fully associative TLB, processor control and status registers, and an on–chip cache controller featuring 267MBytes/second of processor bandwidth using industry standard static RAMs.

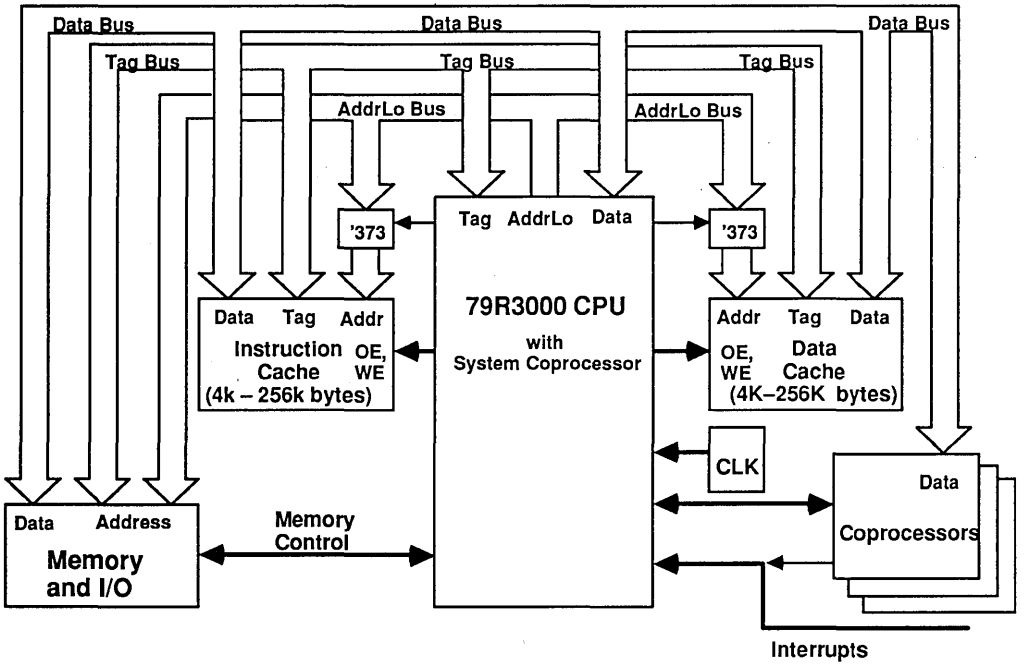


Figure 1. Typical 79R3000 Based System

9

The architecture features a separate floating point accelerator, the 79R3010, which executes at 9.3 single precision Linpack MFLOPS (4.3 MFlops double precision). Also frequently used is the 79R3020 write buffer, which allows the processor to execute stores to main memory at the processor cycle rate, rather than the main memory rate. Figure 1 illustrates a typical 79R3000 based–system.

The full performance of the 79R3000 family is a result of balanced integration between software and hardware. The integer CPU contains a five–stage pipeline, therefore executing five instructions concurrently and reducing the average time per instruction. Optimizing compiler technology serves to both reduce the number of dynamic instructions required to execute a given task, and also to insure that the various resources of the CPU are fully utilized. Examples of compiler optimization include scheduling instructions to eliminate latency effects in branch or load instructions, and also scheduling resources to minimize the occurance of hardware interlocks which might arise if an access to a busy resource is attempted.

ON–CHIP CACHE CONTROL

The 79R3000 contains an on–chip cache controller which controls separate Instruction and Data Caches. Each cache can vary in depth from 4k Bytes through 256kBytes. Additionally, each cache can have independent cache refill block sizes of 1 through 32 words (block refill refers to how much data is retrieved from main memory when processing a cache miss; block refill relies on the principle of locality of reference to improve net processor performance by amortizing the expense of going to slower main memory over a number of instruction or data elements likely to be needed).

The on–chip cache controller implements separate, direct–mapped Instruction and Data caches using industry standard static RAM devices such as the IDT 7198 (16k x 4) or IDT 71586 (4k x 16 with integrated address latch), using a single address and single data bus. External latches capture the address for the cache access, while the 79R3000 provides signals to directly control the address latching, RAM output enable, and RAM write control for each cache. The tag comparison occurs on–chip, simultaneous with the data access, minimizing the amount of time necessary for the cache control function. Thus, it is very simple to implement a highly–efficient cache for the 79R3000 while minimizing the amount of logic needed to implement the cache control. Figure 2 illustrates the design of 64kBytes each of Instruction and Data Cache, a configuration common in workstation applications.

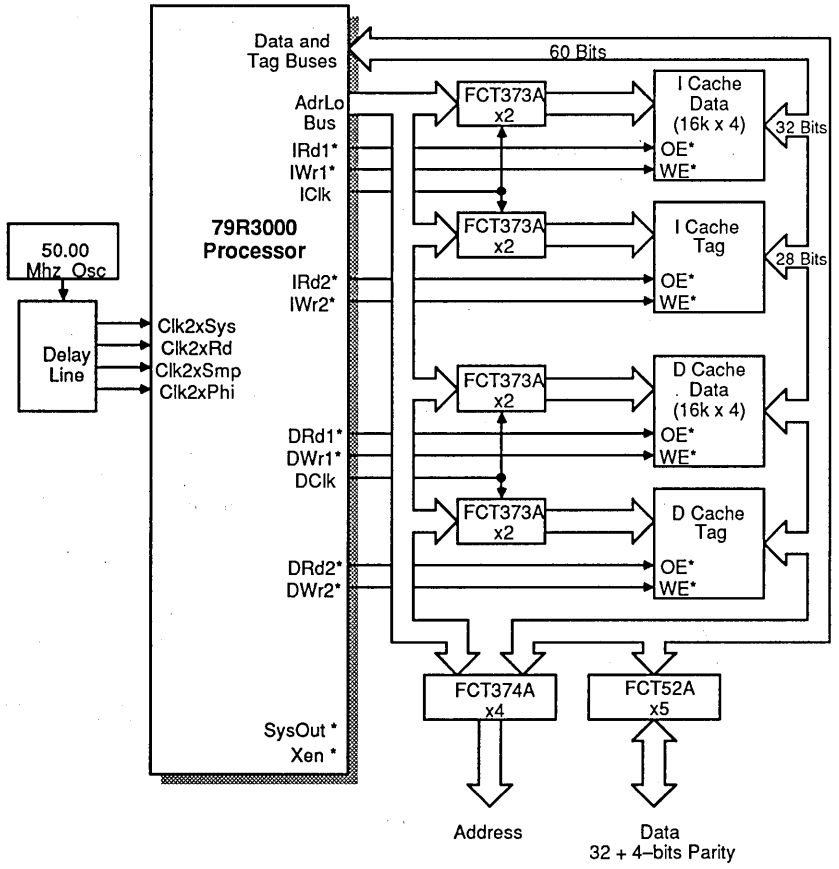


Figure 2. 64kBytes of Instruction and Data Cache

ASYNCHRONOUS MEMORY INTERFACE

The 79R3000 also incorporates a simple, flexible interface to main memory resources, including RAM, PROM, and I/O. This interface is supported through the use of the Asynchronous Memory Interface, which is used whenever uncached data is required (e.g. cache miss processing, uncacheable memory, I/O devices, etc.).

The asynchronous interface consists of control signals for reading and writing main memory. The processor asserts its request signal (e.g. MemRd, which indicates a main memory read), and main memory throttles the processor with the appropriate control signal (e.g. RdBusy, which says that the required data is not yet available). Further information about the type of transaction is available from the Access Type bus, which indicates the size and cause of the data access. The processor automatically updates the contents of its caches when main memory traffic was initiated by a cache miss.

The 79R3000 can be used with a wide variety of memory interfaces. In compute server type systems, the asynchronous interface is typically used as part of a backplane bus arbiter circuit. The memory required by the processor then typically takes many cycles before the first transaction can be completed, while immediately subsequent transactions take considerably fewer cycles (the 79R3000 block refill mechanism takes advantage of this fact). In embedded systems, the processor typically has much more direct control of its main memory resources, and these signals are used with a simple state machine to facilitate transfer activity between the main memory bus and the CPU–cache bus. Thus, the 79R3000 asynchronous memory interface is flexible enough to satisfy a wide variety of applications.

With all of this flexibility, it is important that the system designer have a rich toolset available to assess the implications of certain design decisions. This toolset must account for the application constraints in terms of cost, size and performance. The IDT 79R3000 is supported by tools which allow system designers to evaluate their proposed system in advance of committing to PC board, and to verify that the system design goals are achieved.

PERFORMANCE CONSIDERATIONS

This architecture results in the highest standard of microprocessor performance. In developing an application for the 79R3000, it is important to understand the fundamental reasons for the processor's high standard of performance, and to understand the implication of design decisions in terms of performance. In this way, the cost of the system can be optimized for the performance level required by the application.

The performance of the 79R3000 is a result of its low average clocks per instruction, and its high memory bandwidth. The pipeline stages allow multiple instructions to be in various stages of processing simultaneously, as shown in figure 3. (Compiler technology serves to insure full use of this parallelism.)

The instruction cache allows a new instruction to be initiated on every processor clock cycle, and the low average instruction latency means that instructions are completed at close to one cycle per instruction. The purpose of the instruction cache is to provide the majority of the instructions at the processor clock rate.

Instruction bandwidth is complemented by data bandwidth. The purpose of the system write buffers (such as the IDT 79R3020) is to allow the processor to finish with a data write in one clock cycle, even if the system takes multiple clock cycles to retire the data. The data cache provides for very fast data load operations, and is capable of supplying a data operand in every clock cycle (the 79R3000 can get both an instruction and data item in each clock cycle, resulting in 267MBytes/sec bandwidth at 33Mhz).

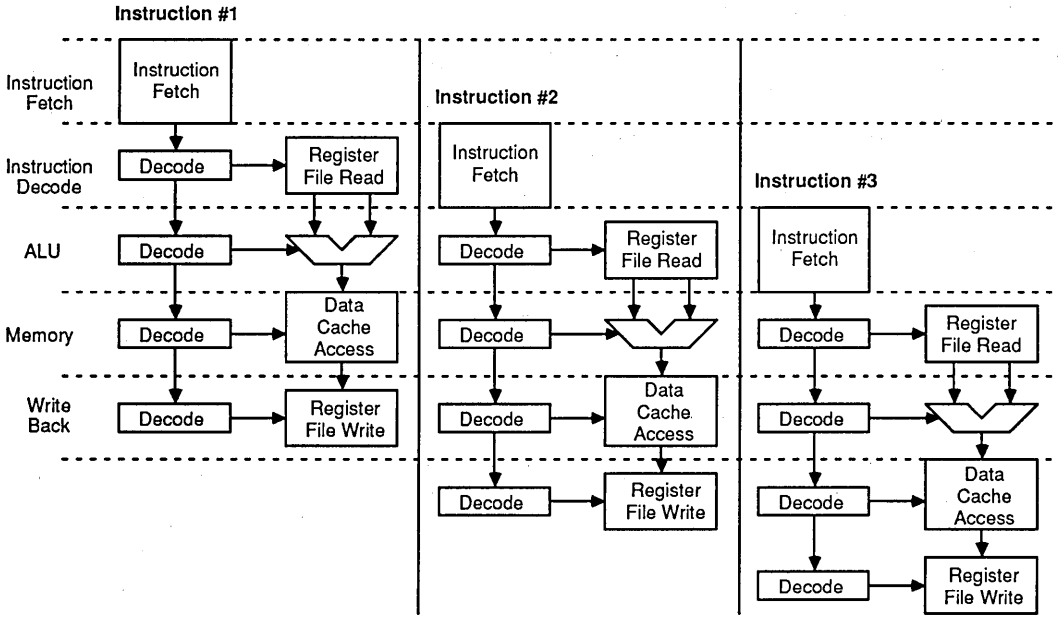


Figure 3. Pipeline Stages of the 79R3000

Performance in a given application is thus to a first approximation a function of system speed and cache efficiency. That is, more cache hits imply that more information is given to the processor at the single cycle rate. Coupled with high frequency (short cycle time) operation, the 79R3000 achieves ultra–high performance. The 79R3000 includes features which further help to minimize the impact of cache miss processing. These features include the block refill capability, which allows multiple main memory words to be retrieved when performing cache miss processing. Block refill takes advantage of the difference between memory latency (the time to the first operand), and memory bandwidth (the time between successive operands). The 79R3000 allows the designer to select the block refill appropriate to the application. Further, the 79R3000 performs instruction streaming, which is the simultaneous fetch and execution of cached instructions, to result in substantial performance improvement when filling the cache with a new process or task.

The amount of cache "sufficient" for a given application is a function of a number of factors. These factors include the latency to main memory (how long on average it takes to receive an operand not in the cache), as well as the instruction mix (the reference and time locality of the code, the amount of data versus ALU operations, etc.). The system designer is free to implement more or less cache, as appropriate for the system, depending on his cost and performance constraints.

One tool available to help the system designer determine the appropriate amount of cache for a given application is the Cache–2000 cache simulator. With Cache–2000, the designer can "describe" a proposed system and evaluate the performance of representative software on that system. Cache–2000 does not require the software engineer to modify source code, but rather works from the executable image of the code using UNIX profiling tools to determine the number of cycles required to execute the program on the described system. Inputs to Cache–2000 include the cache sizes, processor frequency, main memory latency, write buffer depth, and block refill sizes selected. Cache–2000 provides a highly accurate simulation of the software performance on the system described, and gives the system designer a firm basis for making design trade–offs.

A final consideration exists for real time systems, where a general purpose cache may not satisfy the predictability requirements of the application. A number of techniques for the 79R3000 exist which guarantee predictable, high–performance in real time applications. These techniques allow the system designer to use either hardware or software to lock time critical portions of the operating system into cache at all times, guaranteeing single cycle access to kernel instructions. Alternately, the system designer could take advantage of the on–chip cache controller to construct a memory system entirely with synchronous memory devices (e.g. SRAM), and achieve the highest levels of system performance and true predictability.

Figure 4 illustrates alternative ways in which the 79R3000 flexibility and performance can be brought to embedded applications. The performance evaluation tools allow the system designer to conclude in advance whether a given system architecture is appropriate to the task .

COST TRADEOFFS

Once the appropriate system configuration is determined, the system designer can go about designing the lowest cost system achieving the desired level of performance. IDT has written applications notes describing ways to reduce system cost while not affecting application performance. These techniques take advantage of the nature of various types of applications, and use the integration of the 79R3000 to full advantage. By taking advantage of these features, the system designer can build "just as much" cache as is needed for the application.

Specifically, the 79R3000 cache controller was designed to implement a line size of one word (that is, one tag, or main memory origin descriptor per 32–bit word in the cache). Further, the cache controller was designed to support a full 4 Gigabytes of cacheable main memory, and to implement caches between 4k bytes and 256k Bytes in depth. In doing so, the on–chip cache controller will compare 20–bits of tag on each cache cycle.

However, it is not required that all tag values come from ram elements. In systems which implement caches larger than 4k Bytes, it is possible to provide a feedback path between the processor output cache address and the input low–order tags. This can be done with a single 74FCT244A buffer/driver, rather than RAM.

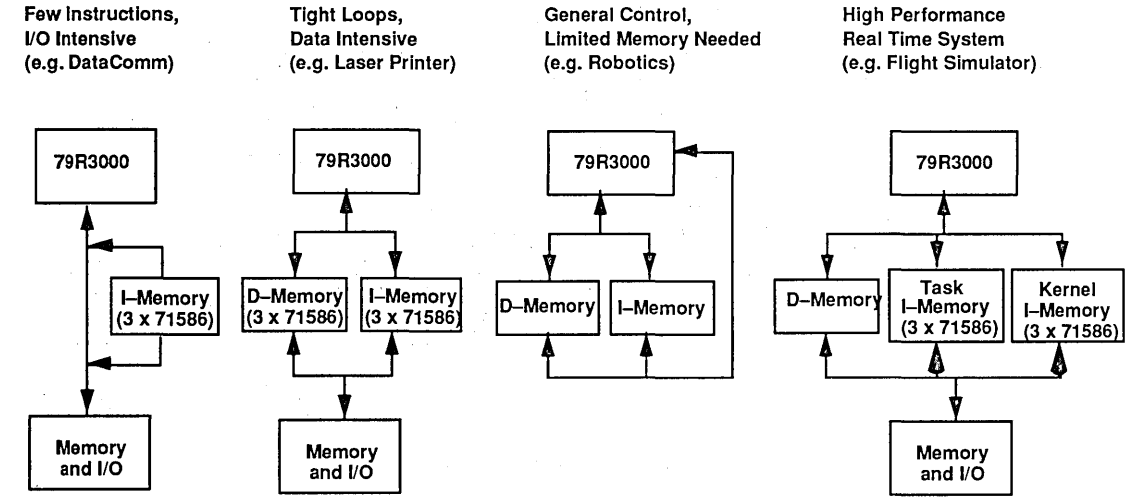


Figure 4. Typical 79R3000–based Systems

Similarly, many embedded applications require less than 4 Gigabytes of cacheable memory (a number more typically associated with reprogrammable applications). In these systems, the designer can "hard–wire" high–order tags, using resistors or buffers, and reduce the cost of the cache by reducing the cacheable memory space. This still allows the system designer to access the entire 4 GigaByte address space for memory decoding, etc.

Additionally, there is no absolute requirement that the cache line size be one word. The system designer could take advantage of the processor's built in block refill capability to implement a larger line size. If the line size were increased to only four words, then one–fourth as many tags would be required as data elements, and the cost of the cache is further reduced.

THE ROLE OF DEVELOPMENT TOOLS

For the system designer to properly design the application, it is important that sufficient tools be in place. Many of these decisions will be determined by the software to be run on the application. The 79R3000 features a wide variety of software development tools, which allow the concurrent development of both hardware and software. These tools allow the software to be developed before the target hardware is designed. Thus, the system designer works from accurate information regarding the nature of the software, its dynamic execution profile, and the amount of memory required to support it. The software developer develops his application on the development host, and uses Sable, the architecture and instruction set simulator, to develop all of the application code prior to target system availability.

The hardware designer can also draw on a variety of development and debug tools. In addition to the Cache–2000 program, there are a number of other tools used by the hardware designer. These include hardware models of the component family, such as those available from Mentor or Valid Logic. These models use actual components to simulate the interaction of the CPU in the target system, while the system exists only in schematic form on the CAD platform. This reduces the risk that the first PC board has significant errors.

When the system is being initially debugged, logic analyzer tools are used to trace the system activity. The 79R3000 is supported on a number of logic analyzers, including Gould, Tektronix, and HP family offerings. These analyzers disassemble the bus traffic, and are invaluable, non–intrusive aids during the debug process.

Software tools are also used during system debug. The PROM monitor, available from IDT, includes useful debug aids such as breakpoint, memory examine and modify, diagnostics, etc. The PROM monitor is available in source form, allowing full customization to the requirements of the target system. Thus, the interaction of the target software and hardware can be examined and confirmed during the debug stages.

PRE–PACKAGED SOLUTIONS

There are a number of other products available to minimize the amount of time required for RISC development. These include hardware and software products.

For the hardware designer, IDT offers a family of RISC subsystems, integrating the CPU and Caches onto a single, small footprint module such as the subsystem shown in figure 5. These modules are ideal for initial development and even for full production. The risk associated with high–speed system design is eliminated, and the customer can focus on his value added software and peripherals.

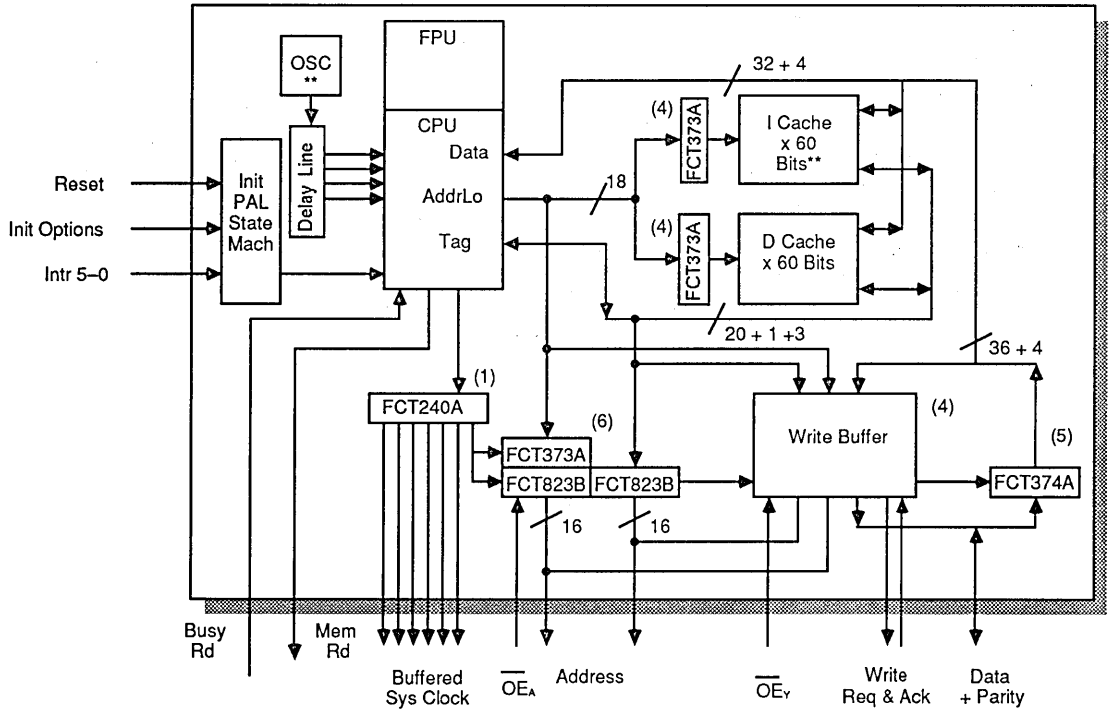


Figure 5. IDT RISC Subsystem

9

For the software designer, a rich toolset is also available. The Stand–alone I/O library interfaces between C language I/O calls and direct device drivers. The PROM monitor eliminates the need for the software developer to develop tools, and thus allows him to focus on the software for the application itself. Finally, a wide variety of operating system support is available, including real time operating systems for either C or Ada applications.

CONCLUSIONS

The IDT79R3000 architecture is ideal for a wide variety of high–performance applications, ranging from embedded applications through real–time systems and including high–performance reprogrammable applications such as workstations and file servers. In addition to its inherent high–performance, the 79R3000 architecture is flexible enough to allow the system designer to configure the system according to the application requirements, thus achieving the desired performance at minimum cost.

In order to bring the performance of the processor to the application system, it is important that a rich and robust toolset be available at all stages of the system development cycle. The 79R3000 provides tools for both the hardware and software teams to use during all phases of the development project, as shown in figure 6.

The end result is the highest–performance processor available today, and a toolset which enables application developers to bring that performance to their systems at minimum cost and with minimal development effort.

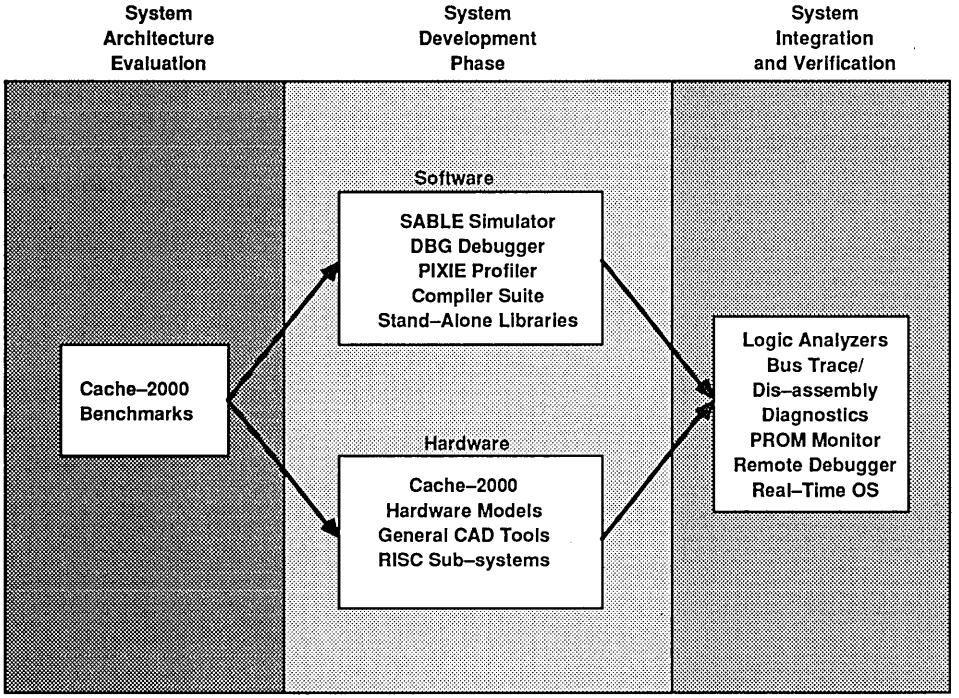


Figure 6. System Development Phases and Support

UNIX is a trademark of AT&T
VAX is a trademark of Digital Equipment Corporation
Macintosh is a trademark of Apple Computer
MIPS and RISC/os are trademarks of MIPS Computer Systems

Integrated Device Technology, Inc.

IDT's R3001 SIMPLIFIES DESIGN OF HIGH-PERFORMANCE CONTROL SYSTEMS

CONFERENCE PAPER CP-03

by Philip Bourekas

ABSTRACT

This paper discusses the architecture of the R3001 RISController™, a derivative of MIPS Computer Systems R3000 microprocessor which IDT has developed to address the particular needs of embedded systems designers. This paper discusses the architectural features of the MIPS RISC architecture which make it well suited to embedded applications, and discusses the changes to the R3000 implementation embodied in the R3001. The paper also gives examples of how these changes help embedded system designers achieve the goals of their applications.

INTRODUCTION

The IDT R3000 microprocessor (the MIPS RISC Processor) has found widespread acceptance among re-programmable applications such as UNIX™ workstations and server systems. Less widely known, but of considerable significance, is the growing acceptance in the embedded marketplace in applications including real-time control systems, data communications, laser printer controllers, graphics terminals, and avionics controllers.

This acceptance is testimony to the elegance of the instruction set, software, and the basic device implementation. In working with such a diversity of customers, however, IDT identified a set of significant changes to the R3000 device which make it even better suited to solving these types of control problems.

While the R3000 is obviously a good device for many embedded applications, it became clear from our customers that it would be possible to implement incremental changes to the device to simplify and broaden its application in embedded systems. IDT has implemented a derivative of the R3000 which addresses many of the issues the R3000 brought to embedded system designers.

The R3001 is the solution. The design goals of the R3001 were to:

- Maintain FULL software compatibility with the R3000, at both the kernel and user levels, to maximize the wealth of software support (both development and applications) available for the MIPS RISC architecture.
- Allow embedded system designers to realize the performance of the R3000 at lower total system cost (fewer devices, less power, less board area, etc.).
- Allow the system designer more options in the design and partitioning of the high-performance memory system.
- Give the system designer full control of all aspects of system design; don't make each system pay for the full set of worst case assumptions about systems dramatically different from his or her target application.
- Recognize the needs of real-time deterministic systems, and provide solutions to the problems of general purpose caches in these applications.
- Support systems which do not wish to implement "cache", such as real-time systems.
- Support the use of complementary products designed for the R3000, such as the high-performance R3010 Floating Point, cache RAMs, interface chips, etc.
- Support systems which distribute the processing task among tightly-coupled heterogenous processing devices, such as systems with I/O processors working directly with the R3001 local memory.
- Maintain same pin count as R3000 (which is pad limited; higher pin count would thus result in higher end-user device cost). Pin compatibility was not a constraint.
- Achieve the same range of speeds as the standard R3000.
- Maintain the full performance of the R3000.

This paper discusses the architecture of the R3001, and how it achieves the above goals. A few design examples are included to help clarify how these changes achieve the goals established for the project.

9

MIPS is a trademark of MIPS Computer Systems, Inc.
VAX is a trademark of Digital Equipment Corp.
UNIX is a registered trademark of AT&T.
RISController™ is a trademark of Integrated Device Technology, Inc.

THE R3000 IN EMBEDDED SYSTEMS

There are a number of reasons why the R3000 has been used in embedded systems. The primary reasons are:

- The R3000 attains the highest levels of performance, based on its efficient architecture: 28 VAX Units of Performance at 33 MHz, sustained, in real-world application. Other processor architectures require much higher frequencies to approximate the performance achieved by the R3000.
- The R3010 Floating Point Accelerator co-processor brings excellent floating point performance to those systems that need it, such as military avionics control systems.
- The compilers for the R3000 achieve most of the efficiency of programming in assembly language, while offering the ease of development of high-level languages.
- It is possible to use a single basic CPU subsystem design, and achieve various levels of performance by either varying cache depth, system speed, or memory interface (write buffer depth and block refill size), and by using hardware or software floating point.
- The R3000 development environment provides powerful software development tools, including debugging support, system profiling, system performance projection, and target system software simulation. This allows system software to be developed quickly and in parallel with the target hardware, minimizing time-to-market.

THE R3000 ARCHITECTURE

The R3000 implements a strictly hierarchical view of memory, appropriate for minicomputer systems. In a typical R3000 system, a given level of memory is a high-speed cache of the larger, slower memory below in the hierarchy, as shown in Figure 1. For example, the on-chip register file contains the most frequently used data items; the next high-speed memory is implemented as high-speed instruction and data caches for the main memory, which is a "cache" of the mass storage system. The R3000 utilizes various techniques to manage the interaction of the various levels, from register allocation algorithms implemented in the compilers to TAG comparison of the cache, and an MMU to support a demand paged virtual memory system.

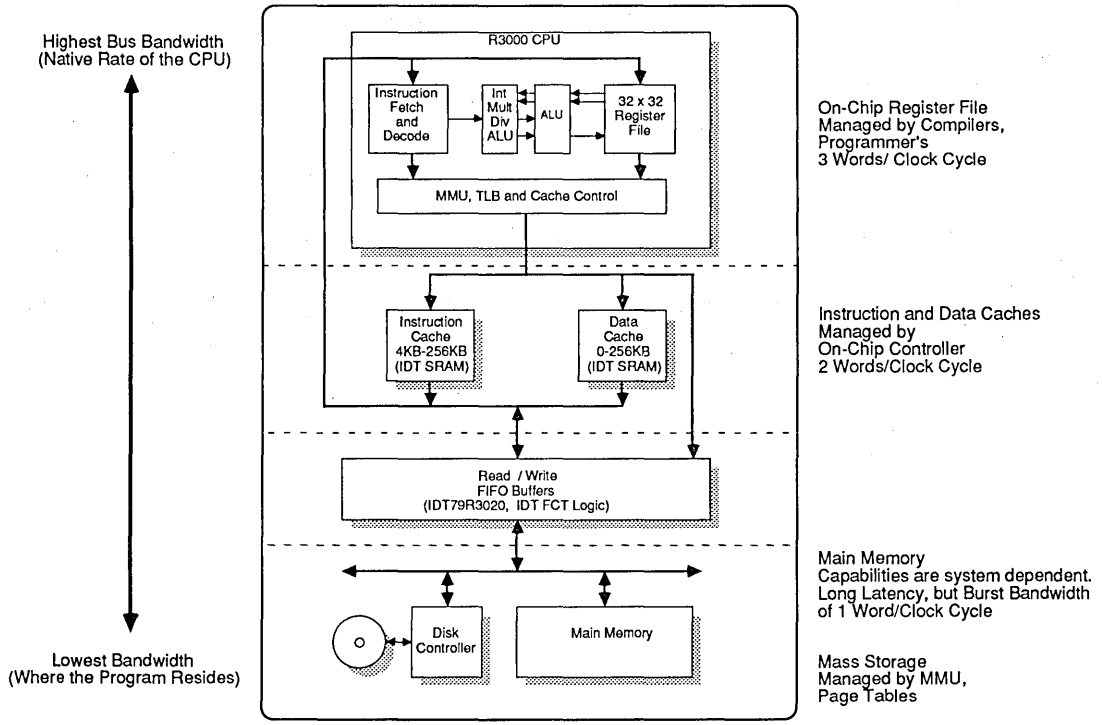


Figure 1. Hierarchies of Memory in an R3000-Based System

The R3000 integrates a direct mapped cache controller on chip. The R3000 requires that a full 60 bits of data be provided by each cache in each cycle; these 60 bits include 32 bits of data, 4 bits of data parity, 20 bits of TAG, a valid bit, and 3 bits of TAG parity. This amount of overhead is required because the R3000 allows (and assumes) cacheable main memory as large as 4 GB, and a cache as small as 4 KB.

All of these techniques combine to allow the R3000 to bring the highest levels of performance to microprocessor-based systems. However, these techniques are not necessarily appropriate to embedded applications, and thus add complexity and cost to the system design. For example, an embedded system typically contains all instructions in the system PROMs; there is no need to perform demand paging from disks. Similarly, embedded systems may not operate in a multi-tasking model, but rather run a single executive process. This simplifies (or eliminates) the requirement for managing virtual memory references.

In fact, many embedded systems do not wish to deal at all with caches. This may be motivated by cost, area, or system architectural requirements. For example, a real-time system may be vastly complicated by a general purpose cache; the system designer can not be guaranteed what is in the cache when a given system event occurs, and thus the general purpose cache brings no performance gain to the real-time system. In fact, in these real-time systems, general purpose caches are viewed as "non-deterministic", and thus contradict a basic requirement of real-time applications.

THE IDT 79R3001 ARCHITECTURE

From the perspective of a programmer, there is no difference between the standard R3000 and the IDT 79R3001. Both devices contain the same basic execution core and memory management unit, thus eliminating any risk of software incompatibility. The R3001 is, therefore, fully software compatible with the R3000, at both the kernel and user levels of software. Neither user programs nor kernels need to be modified to use the R3001.

However, the R3001 can present a totally different look to the system designer compared to the model assumed by the R3000. The R3001 was implemented as a set of design changes to the bus-interface of the R3000. These changes were selected to maximize the amount of flexibility the system designer has when implementing the appropriate memory subsystem for his embedded application. Whereas typical uni-processor computer systems tend to have relatively similar memory requirements, embedded applications tend to have dramatically different requirements, depending on the particular problem to be solved.

THE R3001 MEMORY INTERFACE

While the R3001 does support the hierarchical memory structure of the R3000 (at lower system cost), the RISController goes beyond the capabilities of the R3000 by allowing a different view of memory. Rather than assuming that the highest speed memory is always used as a general purpose cache, the R3001 allows the system designer to use this memory space in a wide variety of ways. This flexibility is the key to the R3001 in embedded systems.

The R3001 allows the system designer to view the fast memory as a synchronous memory space, distinct from (rather than a cache of) slower memory. This partitioning is at the discretion of the system designer; in some systems a general purpose cache model might still be used, while in other systems a "cacheless" memory system would be implemented to assure real-time performance.

The R3001 directly controls two areas of memory: the synchronous memory space, and the asynchronous memory space. Synchronous memory is further subdivided into separate instruction and data portions, to supply the highest possible bandwidth to the processor. Unlike the R3000, it is simple for the system designer to implement these memory spaces as two different regions of memory, rather than implement a hierarchical relationship between them. The memory interfaces supported by the R3001 are illustrated in Figure 2.

The R3001 directly controls the synchronous memory spaces using a single data bus and single address bus but separate sets of control pins (one set for instruction and one for data). The control pins control the external de-multiplexing of address (using external transparent latches) and multiplexing of data (using the output enable of the memory devices). The synchronous interface performs both an instruction and data access per clock cycle on alternate phases of the clock. It presents an address for the instruction in one phase and completes the data transaction in the same phase. In the next phase it completes the instruction transaction and initiates a new data transaction. The operation of the synchronous interface is illustrated in Figure 3.

In systems which wish to implement a hierarchical relationship between the high-speed synchronous memory and slower memory, the R3001 simplifies the design of the high-speed caches by making the on-chip direct-mapped cache controller much more flexible. All the system designer needs to do is "widen" the synchronous memory space by connecting standard memory devices to the appropriate TAG lines of the processor. The CPU will automatically manage the cache, detect hits and process misses by accessing the asynchronous memory space.

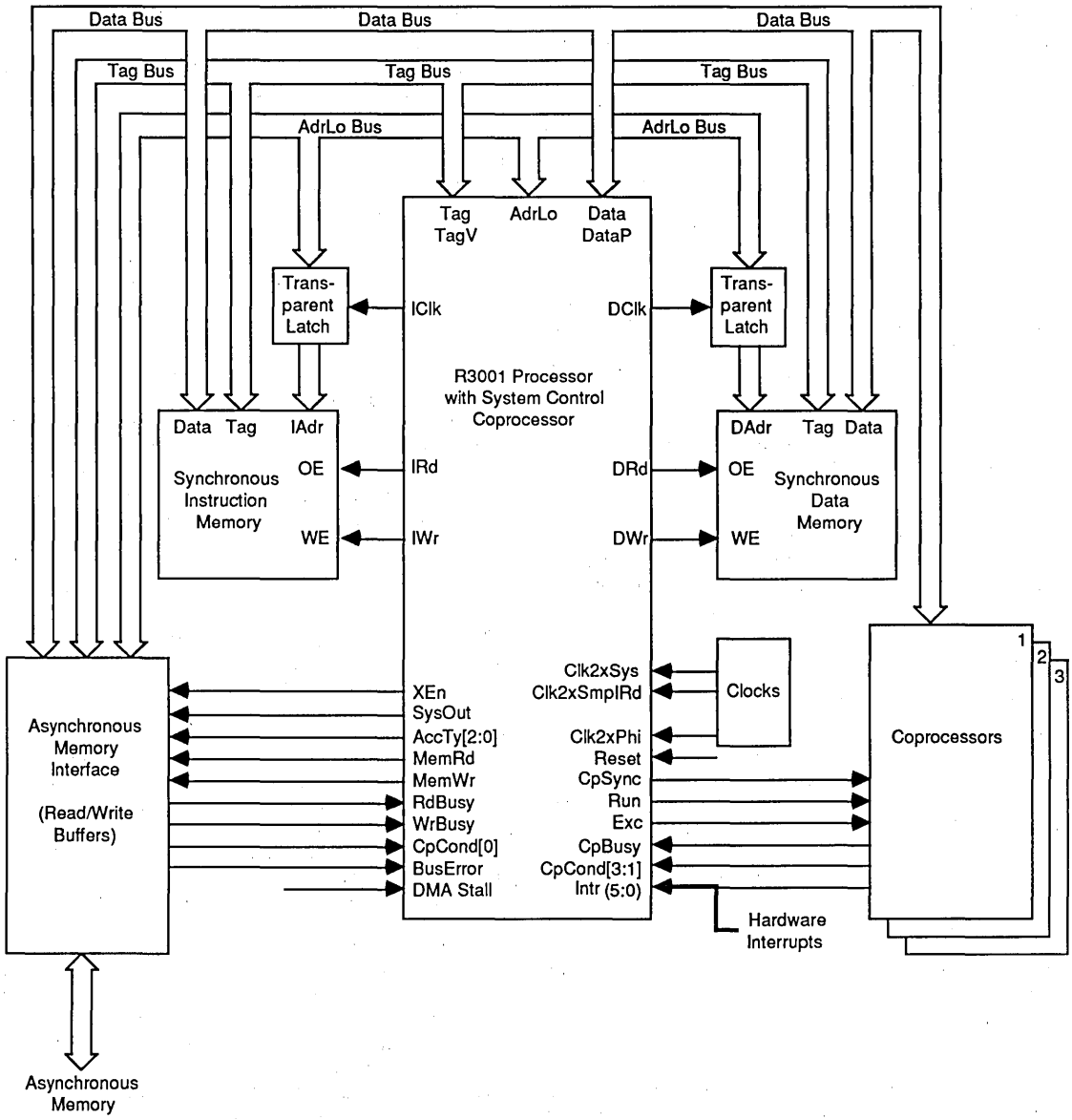


Figure 2. Memory Interfaces of IDT's R3001

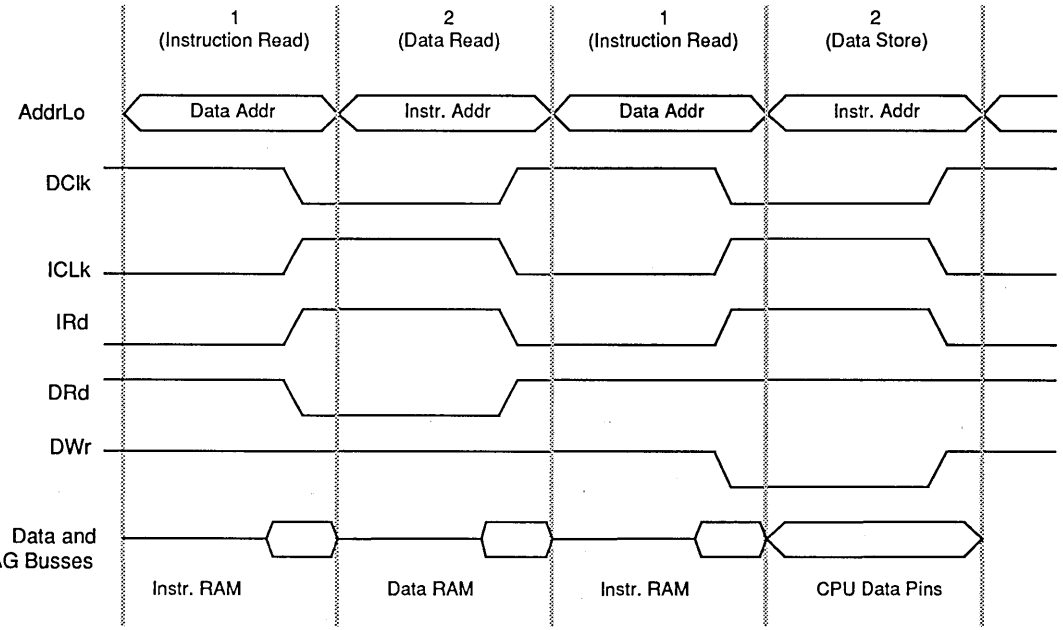


Figure 3. R3001 Synchronous Memory Interface Operation

In these hierarchical systems, the R3001 implements caches with substantially lower cost than the R3000 does. This is because of the flexibility of the synchronous memory interface. The R3001 eliminates TAG parity, and makes data parity optional. Additionally, rather than requiring TAGs for a model of 4 GB memory and a 4 KB cache, the R3001 only requires the amount of TAGs required for that particular system. Most embedded systems implement main memory of 8 MB or less, and build caches of 8 KB or larger. Thus, rather than requiring 60 bits per cache to support the R3000 cache controller (or 15 high-speed 16Kx4 memories), R3001 based systems can be implemented with cache widths of 44 bits or less (only 11 RAMs per cache). This saves a minimum of 8 high-speed RAMs per system, as shown in Figure 4. Note, however, that disabling the checking of TAG bits does not reduce the amount of memory supported by the processor; it reduces the amount of memory which may be *cached.* The full 4 GB address space is still available (although not all of it is cacheable) in the system, simplifying address decoding for various types of memory such as memory-mapped I/O devices.

This same mechanism extends to allow cacheless systems; the system designer merely disables all TAG checking. The architecture of the processor allows software to separate synchronous and asynchronous memory references by using virtual addresses that are either "cacheable" (synchronous) or "uncacheable" (asynchronous).

This approach yields direct benefits to real-time systems. By eliminating the hierarchical model of memory, the high-performance real-time system designer can accurately measure and predict critical real-time metrics such as context switch time and interrupt latency without worrying about the uncertainty associated with general purpose caches.

9

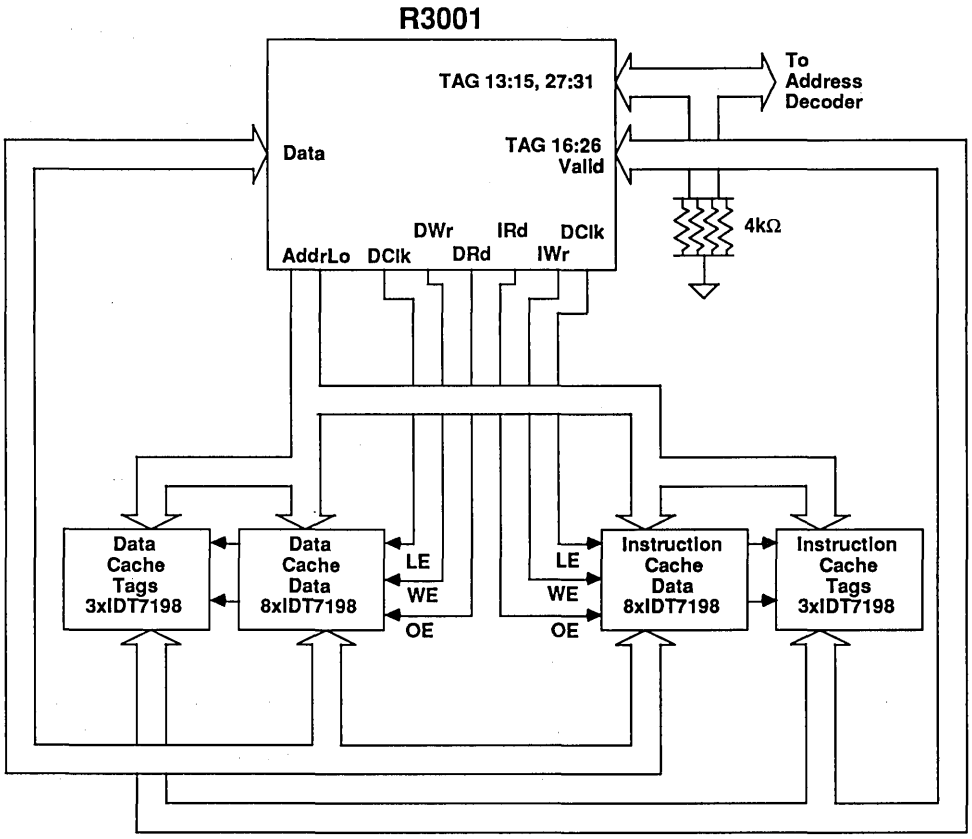


Figure 4. Cost-Reduced R3001 Cached System

DMA SUPPORT

Since the R3000 always assumes a cache-based system, the only external access to the fast memory supported are those required for multi-processor cache coherency. The R3000 thus incorporates a simple but flexible interface which allows an external agent to specify a given cache line, and to request that the processor invalidate that line by writing into it. System DMA events, such as disk transfers, are assumed to only occur in the asynchronous memory space. In this memory space, the processor is not a master, but merely another requester.

While this model is appropriate for minicomputer type systems, it severely limits the embedded system designer. Many embedded systems use different types of specialty processors under the control of a single, general purpose processor. For example, a threat recognition system might contain a DSP subsystem to perform image recognition, and use a general purpose processor for prioritization or response to threats. Such a system needs a high-bandwidth method of communicating between these processors, a natural application for DMA transfers.

The R3001 allows an external master to request mastership of the synchronous memory bus and control signals, as shown in Figure 5. The DMA controller then can quickly transfer data from one processor memory into the high-speed R3001 synchronous memory, allowing the RISController to process the data (or special instruction sequence) rapidly.

CACHELESS SYSTEM

A high-performance processor requires a great deal of memory bandwidth to keep the execution engine running at full speed. Typically, caches are used to supply this bandwidth. In many applications, however, performance can be traded for cost, and the use of caches can be avoided. Alternately, in many other applications, it is feasible to implement a large enough fast memory that traditional caching strategies can be avoided.

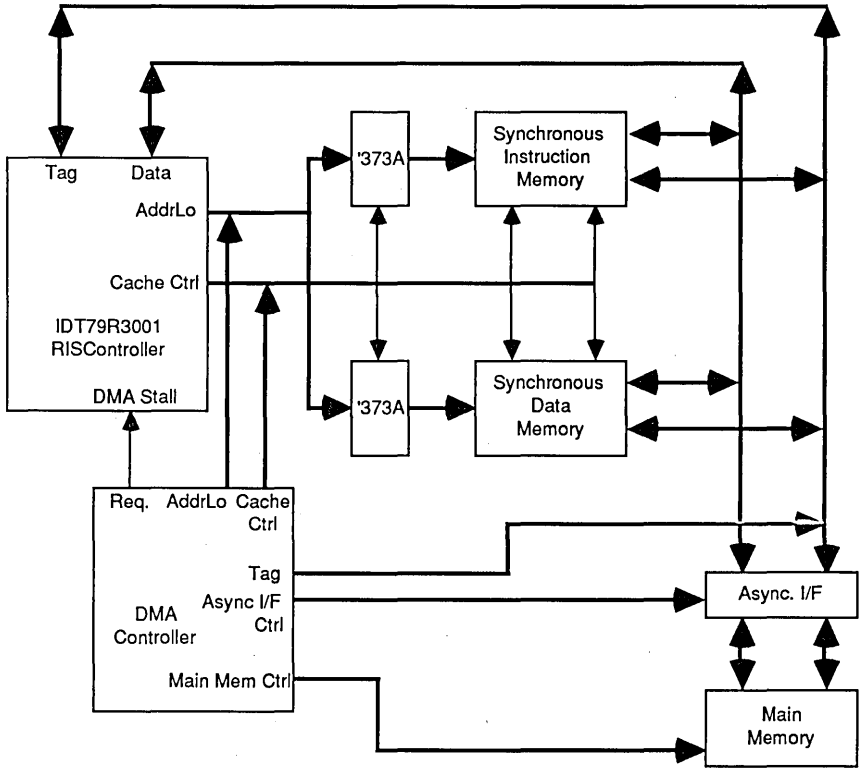


Figure 5. R3001 DMA Arbiter Interface

The system design shown in Figure 6 illustrates a cacheless system offering 7 mips of performance (based on real-world embedded applications such as page-description language interpreters). Rather than using caches, a set of ROMs contains the processor instructions, and the processor clock rate is slowed to 8 MHz to allow a ROM access every clock cycle. In addition to the instruction ROMs in the synchronous space, there is a bank of DRAM for main storage (in a laser printer or graphics application, this would contain the page or screen image and the display lists). Peripheral devices, such as communications devices, are also resident in the asynchronous memory space.

All instruction accesses are satisfied by the ROMs, providing zero-wait state access to instructions. Since the processor clock speed is reduced to 8 MHz, DRAM accesses can be satisfied in 3 clock cycles (even if using 180 ns DRAMs, such as those available in the military market place). Note that a small data cache, implemented using just 5 2Kx8 SRAMs at 90ns (very inexpensive devices) could further increase performance by reducing the number of DRAM references necessary. Implementing the data cache would merely require the inclusion of the address latch and the memory devices.

A similar system could be implemented using dense SRAMs rather than ROMs, allowing higher speed operation. Such a system would probably require the boot ROMs to be resident in the asynchronous memory space. This type of system is typical of many of the real-time applications of the R3001. The R3001 makes this possible because of its processors ability to implement cacheless systems, and because of its support of large (16 MB) synchronous memory.

REAL-TIME CACHE BASED SYSTEM

It is possible to implement a system with the benefits of caching which also satisfies the deterministic requirements of the real-time system designer. With the R3001, the real-time system designer can easily implement a specialized caching structure which guarantees that critical kernel routines will be executed with constant execution time (rather than the variable execution time which is a result of the uncertainty of whether the code is cache resident).

9

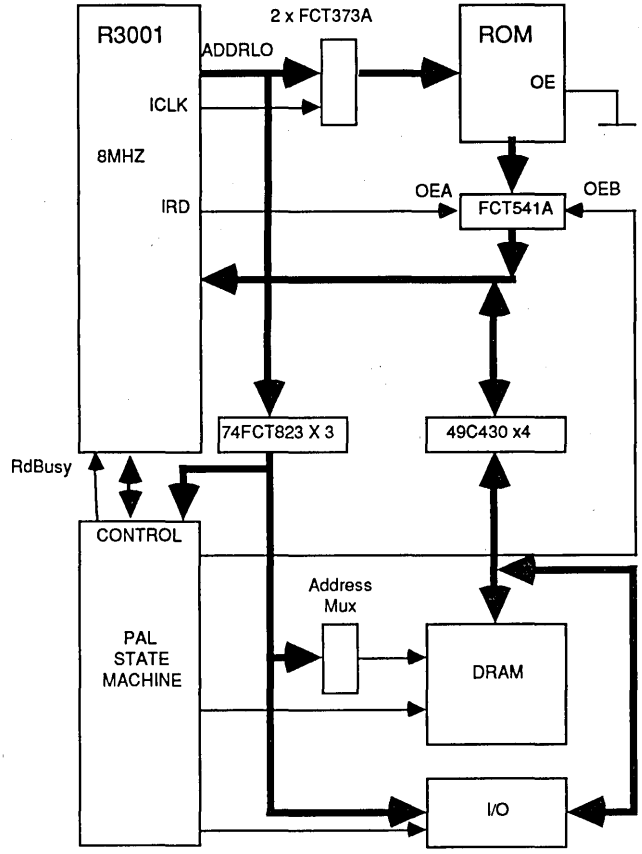


Figure 6. R3001 ROM Based, Cacheless System

The basic technique involves segmenting the cache into two portions: a dedicated kernel section, where the time critical code is resident and which tasks can not overwrite, and a general purpose section, where the tasks benefit from caching but which is non-deterministic.

Software separates memory areas for the time critical code from the general task, according to the addresses used. External address decoding is then used to separate kernel from general accesses, and select the right cache area depending on the decoded address.

Figure 7 shows the cache subsystem which might be implemented for such a design. A high-order address line indicates whether the access is for the time critical kernel, or is a general access. If a kernel access is indicated, then the chip-enable for the kernel cache is activated and the enable for the general cache is negated, thus insuring a kernel reference. A simple routine at system startup pre-loads the critical code into the kernel cache, so that all invocations of those routines result in cache "hits". This achieves the highest levels of real-time performance, utilizes the caches to minimize overall system cost and achieves the highest levels of real-time throughput.

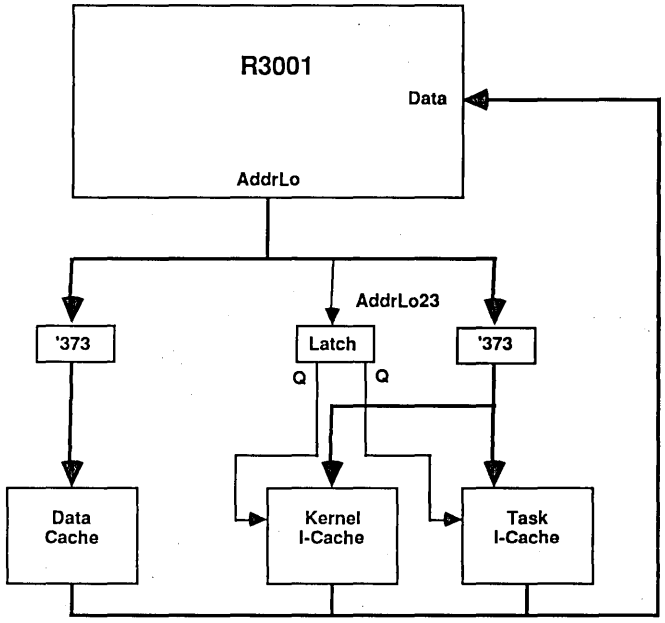


Figure 7. R3001 Real-Time, Cache Based System

SUMMARY

The introduction of the R3001 enables the designers of embedded systems to bring the performance inherent in the MIPS RISC architecture to their embedded applications by satisfying the cost, area, and performance constraints of various applications. This allows the designer to use a single base architecture throughout their system, for example, by using the R3001 to manage I/O processing in an R3000 based minicomputer. This allows both IDT and our customers to leverage their experience with the architecture, and has proven to be attractive to a large number of our customers.

IDT will continue to innovate with the MIPS RISC architecture, producing other derivative versions of the architecture which are well suited to specific embedded applications.

DOMESTIC SALES

ALABAMA

IDT
4930 Corporate Dr., Ste. 1
Huntsville, AL 35805
(205) 721-0211

ALASKA

Westerberg & Associates
Bellevue, WA
(206) 453-8881

ARIZONA

Western High Tech Mktg.
Scottsdale, AZ
(602) 860-2702

ARKANSAS

IDT
(S. Central Regional Office)
14285 Midway Rd., Ste. 100
Dallas, TX 75244
(214) 490-6167

CALIFORNIA

IDT
(Corporate Headquarters)
3236 Scott Blvd.
P.O. Box 58015
Santa Clara, CA
95052-8015
(408) 727-6116

IDT
(Western Headquarters)
2972 Stender Way
Santa Clara, CA 95054
(408) 492-8350

IDT
(SW Regional Office)
6 Jenner Dr., Ste. 100
Irvine, CA 92718
(714) 727-4438

IDT
(SW Regional Office)
16130 Ventura Blvd.,
Ste. 370
Encino, CA 91436
(818) 981-4438

Quest-Rep
San Diego, CA
(619) 565-8797

CANADA (EASTERN)

Canadian Mktg. Tech. Inc.
Kanata, ONT
(613) 591-9555

Canadian Mktg. Tech. Inc.
Mississauga, ONT
(416) 612-0900

Canadian Mktg. Tech. Inc.
Pointe Claire, QUE
(514) 694-6088

CANADA (WESTERN)

Westerberg & Associates
Bellevue, WA
(206) 453-8881

COLORADO

IDT
(NW Regional Office)
1616 17th St., Ste. 370
Denver, CO 80202
(303) 628-5494

Thorson Rocky Mountain
Englewood, CO
(303) 799-3435

CONNECTICUT

Lindco Associates
Woodbury, CT
(203) 266-0728

DELAWARE

IDT
(NE Regional Office)
428 Fourth St., Ste. 6
Annapolis, MD 21403
(301) 858-5423

FLORIDA

IDT
(SE Regional Office)
1413 S. Patrick Dr., Ste. 10
Indian Harbor Beach, FL
32937
(407) 773-3412

IDT
(SE Regional Office)
601 Cleveland St., Ste. 400
Clearwater, FL 34615
(813) 447-2884

IDT
(SE Regional Office)
1500 N.W. 49th St.,
Ste. 500
Ft. Lauderdale, FL 33309
(305) 776-5431

GEORGIA

IDT
(SE Regional Office)
1413 S. Patrick Dr., Ste. 10
Indian Harbor Beach, FL
32937
(407) 773-3412

HAWAII

IDT
(Western Headquarters)
2972 Stender Way
Santa Clara, CA 95054
(408) 492-8350

IDAHO (NORTHERN)

Westerberg & Associates
Bellevue, WA
(206) 453-8881

IDAHO (SOUTHERN)

Westerberg & Associates
Portland, OR
(503) 620-1931

ILLINOIS

IDT
(Central Headquarters)
1375 E. Woodfield Rd.,
Ste. 380
Schaumburg, IL 60173
(708) 517-1262

Synmark Sales
Park Ridge, IL
(708) 390-9696

INDIANA

Arete Sales
Ft. Wayne, IN
(219) 423-1478

Arete Sales
Greenwood, IN
(317) 882-4407

IOWA

Rep Associates
Cedar Rapids, IA
(319) 373-0152

KANSAS

Rush & West Associates
Olathe, KS
(913) 764-2700

KENTUCKY

Norm Case Associates
Rocky River, OH
(216) 333-0400

LOUISIANA

IDT
(S. Central Regional Office)
14285 Midway Rd., Ste. 100
Dallas, TX 75244
(214) 490-6167

MAINE

IDT
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Pkwy.,
Ste. 4002
Westboro, MA 01581
(508) 898-9266

MARYLAND

IDT
(NE Regional Office)
428 Fourth St., Ste. 6
Annapolis, MD 21403
(301) 858-5423

MASSACHUSETTS

IDT
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Pkwy.,
Ste. 4002
Westboro, MA 01581
(508) 898-9266

MICHIGAN

Tritech Sales
Farmington Hills, MI
(313) 442-1200

MINNESOTA

OHMS Technology Inc.
Edina, MI
(612) 932-2920

MISSISSIPPI

IDT
(SE Regional Office)
1413 S. Patrick Dr., Ste. 10
Indian Harbor Beach, FL
32937
(407) 773-3412

MISSOURI

Rush & West Associates
St. Louis, MO
(314) 965-3322

MONTANA

Thorson Rocky Mountain
Englewood, CO
(303) 799-3435

NEBRASKA

IDT
(Central Headquarters)
1375 E. Woodfield Rd.,
Ste. 380
Schaumburg, IL 60173
(708) 517-1262

NEVADA (NORTHERN)

IDT
(Western Headquarters)
2972 Stender Way
Santa Clara, CA 95054
(408) 492-8350

NEVADA (SOUTHERN)

Western High Tech Mktg.
(Clark County, NV)
Scottsdale, AZ
(602) 860-2702

NEW HAMPSHIRE

IDT
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Pkwy.,
Ste. 4002
Westboro, MA 01581
(508) 898-9266

NEW JERSEY

IDT
(NE Regional Office)
One Greentree Centre,
Ste. 202
Marlton, NJ 08053
(609) 596-8668

SJ Associates
Mt. Laurel, NJ
(609) 866-1234

NEW MEXICO

Western High Tech Mktg.
Albuquerque, NM
(505) 884-2256

NEW YORK

IDT
(NE Regional Office)
250 Mill St., Ste.107
Rochester, NY 14614
(716) 546-4880

Quality Components
Buffalo, NY
(716) 837-5430

Quality Components
Manlius, NY
(315) 682-8885

SJ Associates
Rockville Centre, NY
(516) 536-4242

NORTH CAROLINA

Tingen Technical Sales
Raleigh, NC
(919) 870-6670

OHIO

Norm Case Associates
Rocky River, OH
(216) 333-0400

OKLAHOMA

IDT
(Central Headquarters)
1375 E. Woodfield Rd.,
Ste. 380
Schaumburg, IL 60173
(708) 517-1262

OREGON

Westerberg & Associates
Portland, OR
(503) 620-1931

PENNSYLVANIA (WESTERN)

Norm Case Associates
Rocky River, OH
(216) 333-0400

PENNSYLVANIA (EASTERN)

SJ Associates
Rockville Centre, NY
(516) 536-4242

RHODE ISLAND

IDT
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Pkwy.,
Ste. 4002
Westboro, MA 01581
(508) 898-9266

SOUTH CAROLINA

IDT
(SE Regional Office)
1413 S. Patrick Dr., Ste. 10
Indian Harbor Beach, FL 32937
(407) 773-3412

TEXAS

IDT
(S. Central Regional Office)
6034 W. Courtyard Dr.,
Ste. 305-48
Austin, TX 78730
(512) 338-2440

IDT
(S. Central Regional Office)
14285 Midway Rd., Ste. 100
Dallas, TX 75244
(214) 490-6167

UTAH

Anderson Associates
Bountiful, UT
(801) 292-8991

VERMONT

IDT
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Pkwy.,
Ste. 4002
Westboro, MA 01581
(508) 898-9266

VIRGINIA

IDT
(NE Regional Office)
428 Fourth St., Ste. 6
Annapolis, MD 21403
(301) 858-5423

WASHINGTON

Westerberg & Associates
Bellevue, WA
(206) 453-8881

IDT
(NW Regional Office)
7981 168th Ave. N.E.,
Ste. 32
Redmond, WA 98052
(206) 881-5966

WEST VIRGINIA

Norm Case Associates
Rocky River, OH
(216) 333-0400

WISCONSIN

Synmark Sales
Park Ridge, IL
(708) 390-9696

WYOMING

Thorson Rocky Mountain
Englewood, CO
(303) 799-3435

IDT TECHNICAL CENTERS

Integrated Device Technology
(Western Headquarters)
2972 Stender Way
Santa Clara, CA 95054
(408) 492-8350

Integrated Device Technology
(Southwestern Regional Office)
6 Jenner Drive, Suite 100
Irvine, CA 92718
(714) 727-4438

Integrated Device Technology
(South Central Regional Office)
14285 Midway Road, Suite 100
Dallas, TX 75244
(214) 490-6167

Integrated Device Technology
(Eastern Headquarters)
#2 Westboro Business Park
200 Friberg Parkway, Suite 4002
Westboro, MA 01581
(508) 898-9266

INTEGRATED DEVICE TECHNOLOGY, INC.
(European Headquarters/Northern Europe Regional Office)
21 The Crescent
Leatherhead
Surrey, UK KT228DY
Tel.: 44-372-377375

AUTHORIZED DISTRIBUTORS

HALL-MARK ELECTRONICS | **HAMILTON/AVNET** | **INSIGHT ELECTRONICS** | **VANTAGE COMPONENTS** | **ZENTRONICS**

Contact your local office.

INTERNATIONAL SALES

AUSTRALIA

George Brown Group
Rydalmere, Australia
Tel.: 612-638-1999

George Brown Group
Hilton, Australia
Tel.: 618-352-2222

George Brown Group
Blackburn, Australia
Tel.: 613-878-8111

AUSTRIA

Ing. Erst. Steiner
Vienna, Austria
Tel.: 43-222-827-4740

BELGIUM

Betea S.A.
Sint-Stevens-Wolnne,
Belgium
Tel.: 323-736-1080

DENMARK

Exatec A/S
Copenhagen, Denmark
Tel.: 45-31-191022

FEDERAL REPUBLIC OF GERMANY

IDT
(Central Europe Regional Office)
Gottfried-Von-Cramm-Str. 1
8056 Neufahrn
Federal Repulic of Germany
Tel.: 49-8165-5024

Dacom GmbH
Stuttgart, FRG
Tel.: 49-711-780-6810

Dacom GmbH
Ismaning, FRG
Tel.: 49-89-964-880

Dacom GmbH
Buxheim, FRG
Tel.: 49-08-458-4003

Dacom GmbH
Soligen, FRG
Tel.: 49-21-259-3011

Dacom GmbH
Karlsruhe, FRG
Tel.: 49-72-14-7193

Dacom GmbH
Sarstedt, FRG
49-89-5066-5160

Scantec GmbH
Planegg, FRG
Tel.: 49-89-859-8021

Scantec GmbH
Kirchheim, FRG
Tel.: 49-89-70-215-4027

Scantec GmbH
Ruckersdorf, FRG
Tel.: 49-89-91-157-9529

Topas Electronic GmbH
Hannover, FRG
Tel.: 49-51-113-1217

Topas Electronc GmgH
Quickborn, FRG
Tel.: 49-4106-73097

FINLAND

Comodo Oy
Helsinki, Finland
Tel.: 358-0757-2266

FRANCE

IDT
(Southern Europe Regional Office)
15 Rue du Buisson aux Fraises
91300 Massy, France
Tel.: 33-1-69-30-89-00

Scientec REA
Chatillon, France
Tel.: 33-149-652750

Scientec REA
Cesson-Sevigne, France
Tel.: 33-99-32-1544

Scientec REA
Saint Etienne, France
Tel.: 33-77-79-7970

Scientec REA
Venissieux, France
Tel.: 33-78-00-0415

Scientec REA
Cedex, France
Tel.: 33-61-39-0989

A2M
BUC, France
Tel.: 33-39-54-9113

Aquitech
Merignac, France
Tel.: 33-56-55-1830

Aquitech
Cedex, France
Tel.: 33-1-40-96-9494

Aquitech
Rennes, France
Tel.: 33-99-78-3132

Aquitech
Lyon, France
Tel.: 33-72-73-2412

HONG KONG

IDT
(Hong Kong Regional Office)
Unit 329, 3/F Asia Business Centre
The Centre Mark,
287-299 Queen's Road Central
Hong Kong
Tel.: 852-542-0067

Lestina International Ltd.
Kowloon, Hong Kong
Tel.: 852-735-1736

INDIA

Malhar Corp.
Rosemont, PA
Tel.: 215-527-5020

ISRAEL

Vectronics, Ltd.
Herzlia, Israel
Tel.: 972-52-556070

ITALY

Lasi Electronica
Milano, Italy
Tel.: 39-66-101370

Microelit SRL
Milan, Italy
Tel.: 39-2-469044

Microelit SRL
Rome, Italy
Tel.: 39-6-8894323

JAPAN

IDT
(Japan Headquarters)
U.S. Bldg. 201
1-6-15 Hirakarasho,
Chiyoda-Ku
Tokyo 102, Japan
Tel.: 81-3-221-9821

Dia Semicon Systems
Tokyo, Japan
Tel.: 81-3-439-2700

Kanematsu Semiconductor Corp.
Tokyo, Japan
Tel.: 81-3-511-7791

Marubun
Tokyo, Japan
Tel.: 81-3-639-9897

NKK Corp.
Tokyo, Japan
Tel.: 81-3-228-3826

Tachibana Tectron Co., Ltd.
Tokyo, Japan
Tel.: 81-3-793-1171

KOREA

Eastern Electronics
Seoul, Korea
Tel.: 822-566-0514

NETHERLANDS

Auriema
Eindhoven, Netherlands
Tel.: 31-40-816565

NORWAY

Eltron A/S
Oslo, Norway
Tel.: 47-2-500650

SINGAPORE

Data Source Pte. Ltd.
Lorong, Singapore
Tel.: 65-291-8311

SOUTH AMERICA

Intectra Inc.
Mountain View, CA
Tel.: 415-967-8818

SPAIN

Anatronic, S.A.
Madrid, Spain
Tel.: 34-154-24455

Anatronic, S.A.
Barcelona, Spain
Tel.: 34-3-258-1906

SWEDEN

Svensk Teleindustri AB
Spanga, Sweden
Tel.: 46-8-761-7300

SWITZERLAND

W. Stolz AG
Baden-Daettwil,
Switzerland
Tel.: 41-56-849000

W. Stolz AG
Lausanne, Switzerland
Tel.: 41-21-274838

TAIWAN

Johnson Trading Company
Taipei, Taiwan
Tel.: 886-273-31211

General Industries Inc.
Taipei, Taiwan
Tel.: 886-2764-5126

UNITED KINGDOM

IDT
(European Headquarters/ Northern Europe Regional Office)
21 The Crescent
Leatherhead
Surrey, UK KT228DY
Tel.: 44-372-363339

Micro Call, Ltd.
Thame Oxon, England
Tel.: 44-84-261-939

Integrated
Device Technology, Inc.

3236 Scott Boulevard
Santa Clara, CA 95054-3090
(408) 727-6116 FAX: (408) 492-8674

DBK-RISC-00021